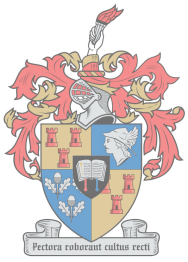
Investigating diffusion on networks with long-range interactions: application to object clustering using generalised heat kernel invariants

by

Alice Nanyanzi



Thesis presented in partial fulfilment of the requirements for the degree of Master of Science in Mathematics in the Faculty of Science at Stellenbosch University

Supervisors:

Prof. Franck Kalala Mutombo
Dr. Simukai Utete

December 2018

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: December 2018 .....

Abstract

Investigating diffusion on networks with long-range interactions: application to object clustering using generalised heat kernel invariants

Alice Nanyanzi

Department of Mathematical Sciences,
University of Stellenbosch,
Private Bag X1, Matieland 7602, South Africa.

Thesis: MSc

December 2018

Diffusion on networks is widely used to model dynamical processes on networks such as propagation of information through a social network for marketing purposes [79], spread of epidemics within a social network [41], among others. The most common model (which we refer to as the standard diffusion model) studied in literature is based on the idea that a diffusing substance spreads from one node to another along the edges of the network. Backed by empirical evidence from real-world processes, Estrada et al. [40] put forward another diffusion model in which the substance is considered to spread not only via the edges but also through long-range interactions between non-nearest nodes. These interactions are accounted for using the Mellin or Laplace transforms of k-path Laplacian matrices, where k is the shortest path distance between a given pair of nodes [38]. We propose to refer to this model as the generalised diffusion model. We study this model in some detail and perform simulations based on this model to ascertain the impact of long-range interactions on the diffusion process on networks of different structures.

The diffusion equation in its simple and generalised version has been extensively studied in the literature [40, 118, 119]. Its fundamental solution, known as the heat kernel, is very vital with applications ranging from graph characterisation and clustering [118], community detection [72] and node centrality measure [24]. Particularly, we focus on the application of the heat kernel for graph clustering purposes. In [118], the heat kernel associated with the standard diffusion equation is studied and the utility of its invariants to graph characterisation for purposes of clustering has been explored in detail. In this thesis, however, we extend this concept by considering the generalised heat kernel, that is to say, the heat kernel associated with diffusion on networks with long-range interactions taken into account. We discuss how

to extract stable and useful invariants of the heat kernel and the use of these invariants in characterisation of graphs for purposes of clustering.

Given a database of objects, we obtain a set of graphs representing each of the images and further show how different invariants of the generalised heat kernel can be used to construct feature vectors used for clustering of the objects. We further investigate the impact of long-range influence on the quality of resulting clustering. From the results of the experiments, we deduce that as the value of exponents of the Mellin and Laplace transforms of k-path Laplacian matrices decreases, better clusters are obtained. This can be explained by the fact that an increase in the value of exponents results in a decrease in strength of long-range interactions.

Uittreksel

Ondersoek diffusie op netwerke met langafstand interaksies: toepassing om klustering te beswaar deur gebruik te maak van algemene hitte kern invariante

Alice Nanyanzi

Departement Wiskundige Wetenskappe,
Universiteit van Stellenbosch,
Privaatsak X1, Matieland 7602, Suid Afrika.

Tesis: MSc

Desember 2018

Diffusie op netwerke word wyd gebruik om dinamiese prosesse op netwerke soos verspreiding van inligting deur 'n sosiale netwerk vir bemarkingsdoeleindes [79], verspreiding van epidemies met in 'n sosiale netwerk [41], onder andere. Die mees algemene model (wat ons verwys na as die standaard diffusie model) bestudeer in die literatuur is gebaseer op die idee dat 'n dif- smeltstof versprei van een knoop na 'n ander langs die kante van die netwerk. Gerugsteun Deur empiriese bewyse uit werklike prosesse, Estrada et al. [40] stel nog 'n ander uiteensetting voor samesmeltingsmodel waarin die stof nie net oor die kante versprei word nie, maar ook deur langafstand interaksies tussen nie-naaste nodes. Hierdie interaksies word verantwoord vir die gebruik van die Mellin- of Laplace-transformasies van k-pad Laplaciese matrikse, waar k die kort- est pad afstand tussen 'n gegewe paar nodes [38]. Ons stel voor om na hierdie model te verwys as die algemene diffusie model. Ons bestudeer hierdie model in detail en voer simulasies uit gebaseer op hierdie model om die impak van langafstandinteraksies op die diffusieproses te bepaal op netwerke van verskillende strukture.

Die diffusievergelyking in sy eenvoudige en algemene weergawe is omvattend bestudeer die literatuur [40, 118, 119]. Die fundamentele oplossing, bekend as die hitte kern, is baie belangrik met toepassings wat wissel van grafiek karakterisering en clustering [118], gemeenskap de- tection [72] en node sentraliteitsmaatreël [24]. In die besonder fokus ons op die toepassing van die hitte kern vir grafiek clustering doeleindes. In [118], word die hittepyp geassosieer met die standaard- Dard diffusievergelyking word bestudeer en die nut van sy invariante om grafiekkarakterisering te bepaal Vir die doeleindes van clustering is in detail ondersoek. In hierdie proefskrif verruim ons dit egter konsep deur die algemene hitte kern te oorweeg, dit wil sê die hittepyp wat verband hou met diffusie op netwerke met langafstand

interaksies in ag geneem. Ons bespreek hoe Om stabiele en bruikbare invariante van die hitte kern en die gebruik van hierdie invariante in te onttrek karakterisering van grafieke vir die doel van clustering.

Gegee ’n databasis van voorwerpe, kry ons ’n stel grafieke wat elk van die beelde verteenwoordig wys verder hoe verskillende invariante van die genormaliseerde hitte kern gebruik kan word om te konstrueer funksie vektore wat gebruik word vir die samestelling van die voorwerpe. Ons ondersoek verder die impak van langtermyn- omvang invloed op die kwaliteit van die gevolglike clustering. Uit die resultate van die eksperimente, ons aflei dit as die waarde van eksponente van die Mellin- en Laplace-transformasies van k-pad Laplaciese matrikse afneem, beter clusters word verkry. Dit kan verklaar word deur die feit dat ’n toename in die waarde van eksponente lei tot ’n afname in sterkte van langafstand interaksies.

Acknowledgements

First, I thank the Lord Almighty for life, wisdom, knowledge and good health throughout the duration of my studies. I would like to express my gratitude to my supervisors Prof. Franck Kalala Mutombo and Dr. Simukai Utete for their guidance, motivation and time. Moreover, I thank my friends and family for the support and encouragement. Last but not least, sincere appreciation to all researchers who tirelessly carry out research and publish their findings thus generously contributing to the existing literature from which knowledge is acquired to further research in the respective areas.

Dedications

To my family and friends.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we briefly introduce the concept of diffusion on networks, the heat kernel and discuss related work. We also present the research question, explain our contributions to this area and finally give a summary of the thesis structure.

1.1 Diffusion on networks

Among many other definitions, Newman [86] defines diffusion as the movement of substance, such as heat, gas, gossip, and information, from a region of high concentration to one of low concentration.
In network theory, diffusion on networks is widely used in developing simple models of processes in the real-world, for example spread of epidemics, dissemination of information within social network, among others. The diffusion process on networks is modelled using diffusion (or heat) equation in which the Laplacian matrix of the corresponding network plays a key role. To start with, researchers have developed various models in which diffusion is considered to occur through interactions along edges of a graph. In other words, there is no interaction between non-nearest neighbours. We choose to refer to this model as the standard diffusion model. This model has been extensively studied with applications to modeling the spread of information in social media marketing [79], propagation of innovation within social networks, and many others.

However, in many real-world processes such as consensus process among agents, synchronisation processes, spread of non-contact epidemics, among others, it was observed that there exist some kinds of interaction between components or agents or individuals that are not directly connected to each other. Such kind of interactions, known as long-range interactions, were not accounted for in the previously discussed case though they contribute significantly to the dynamical processes. A number of experiments aimed at studying and understanding the existence of both direct and long-range interactions in real-world, have been carried out. For example, in the United States of America where an experiment to trace the circulation of bank notes was performed, the tracking mobile phone usage, among others [41].

In these experiments, long-range interactions are considered as random. However, based on empirical evidence, Estrada et al. [41] showed that in modelling the spread of infections in social networks, the long-range interactions are considered non-random and the strength of influence due to these interactions is a function of the shortest distance between pairs of nodes in the network. In [38, 39], Estrada presents a method of accounting for long-range interactions using Laplace and Mellin transforms of the k-path Laplacian matrices. The linear combination of k-path Laplacian matrices for k in the interval $[1, d_{max}]$ (where d_{max} denotes the diameter of the graph), results in a matrix which we refer to as the generalised Laplacian matrix.

In later chapters, we will perform simulations to explore the impact of long-range interactions on networks of different structure, that is, the random graphs, particularly Erdős-Rényi networks, and scale free networks, specifically Barabási-Albert networks.

1.2 The heat kernel

The heat kernel is the fundamental solution of the heat or diffusion equation and it is obtained by exponentiating the spectrum of the Laplacian matrix over time. The heat kernel is a very important aspect of diffusion as it describes the flow of substance across edges within a graph. Many researchers have devoted time to the study of the heat kernel and a lot of literature has been published explaining various applications of the heat kernel for example as a pagerank of a graph specifically in determination of the ranking of web pages by web search engines [24], as a local graph partitioning technique [25], as a community detection tool in networks [72], as a means of graph embedding [119], among others.
A number of computer vision and pattern recognition problems such as graph clustering, pattern matching, image segmentation, routing, sequencing of relational data, among others have been solved using spectral graph theory which involves the use of eigenvalues, eigenvectors and characteristic polynomials of the adjacency or Laplacian matrix to characterise the structural properties of graphs. In this study, we focus on graph clustering which is a very important challenge encountered in computer vision. We approach this problem by use of the heat kernel. According to Xiao [119] different invariants of the heat kernel for example the trace, zeta function, heat content, and derivative of zeta function at the origin, can be used in structural characterisation of graphs for purposes of clustering. This is achieved by extracting feature vectors from the invariants followed by performing Principal Component Analysis (PCA) on the vectors to map the corresponding graphs onto low-dimensional spaces.

1.3 Research question and contributions

The heat kernel associated with diffusion in which substance flows along the paths of a network has been studied extensively with applications in various areas. In particular, Xiao [119] has explored the use of invariants of the heat kernel such as the trace, zeta function, derivative of zeta function at the origin, and heat content for characterisation and clustering of graphs. Following from his work briefly discussed in section 1.2, we pose a research question: Let us consider a diffusion model on a network in which the spread occurs both through direct and long-range interactions. We call such model the generalised diffusion model. Can the invariants such as the trace, zeta function, heat content, derivative of the zeta function at origin, associated with the generalised heat kernel be used for graph characterisation for purposes of clustering? If so, what is the impact of long-range interactions on the quality of clusters?

In this thesis, we contribute to the existing work in this area by introducing the heat kernel resulting from the diffusion model in which both direct and long-range interactions are considered. We propose to call it the generalised heat kernel. We then explore the invariants of this heat kernel and ascertain their utility in graph characterisation and graph clustering in general. Given a database consisting of objects, we extract graphs representing each of the objects. For each extracted graph, we will construct feature vectors based on the invariants above and then perform graph clustering using principal components analysis on the feature vectors. The resulting clusters corresponds to object clusters.

1.4 Thesis structure

The thesis is composed of five chapters and is structured as follows: The current chapter (1) is the introduction in which we discuss brief background and motivation of the heat kernel concept. We further propose the idea of extending the heat kernel concept by including long-range interactions in the diffusion model and utility of its invariants for graph characterisation.

In chapter 2, we review the literature related to this thesis. This includes an introduction to networks and graphs, terminology and definitions in graph theory, matrix representation of graphs, random models of networks, characterisation and structure of networks. It also consists of background on graph similarity and embedding, Voronoi diagrams and triangulations, corner detection in images and finally Principal Component Analysis.

In chapter 3, we explore diffusion on networks where we illustrate diffusion of heat on network using a model where flow of heat occurs along the edges of a network. We also discuss the equilibrium behaviour. We further ascertain the impact of network structure, choice of initial nodes and network homogeneity on the dynamics of diffusion on a network. We also

introduce the idea of diffusion on directed networks in comparison to undirected networks. Moreover, we discuss an extended model where diffusion is not only due to interaction between neighbouring nodes but also through long-range interactions. We refer to this model as the generalised diffusion model. We illustrate this model on networks of different structure, that is, networks with power-law degree distribution (Barabási-Albert networks) and networks with Poisson degree distribution (Erdős-Rényi networks). We further illustrate the impact of long-range interactions on the rate of diffusion on both networks.

In chapter 4, we study the concept of the heat kernel and its utility in graph characterisation. We then extend this concept by considering the generalised diffusion model. In other words, the heat kernel which we refer to as the generalised heat kernel is the fundamental solution of the generalised diffusion equation. We discuss how stable invariants such as the trace, zeta function, derivative of the zeta function and heat content invariants can be extracted from the generalised heat kernel and how these invariants can be used for graph clustering. Experiments are performed on the Columbia Object Image Library (COIL-100) [85] database.

Finally, in chapter 5, we draw conclusions and discuss areas of further research.

Chapter 2

Literature review

2.1 Graphs and networks. An introduction

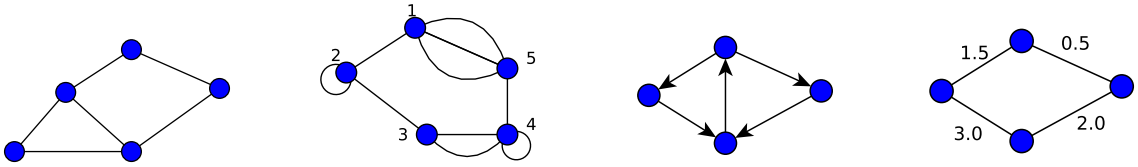
Graph theory is a branch of mathematics that deals with the study of graphs or networks [37]. Most often, the two words: 'graph' and 'network' are used interchangeably, so shall we in this thesis. To begin with, let us discuss some of the terminology that we will use often in this work.

2.1.1 Graph/network terminology

Definition 2.1.2 (Graph)**.** A graph is a pair $G = (V, E)$, where V is a set of vertices or nodes, and E is a set of edges connecting vertices, $E \subseteq \{(u, v)|u, v \in V\}$. A graph may be undirected where edges have no directions or it may be directed (known as digraph) in which each edge has a direction with each edge drawn as a line with an arrow pointing to one of the two vertices that it connects. The order (or size) of a graph G, denoted as $|G|$, is the number of vertices of a graph. On the other hand, the number of edges of a graph is denoted by $\|G\|$. The size of a graph determines whether it is finite or infinite.

A graph with multiple edges is a multi-graph. On the other hand, a simple graph is a graph with neither multiple edges or self loops. Moreover, a weighted graph is a graph in which each edge $e = \{i, j\}$ is associated with a value or weight, $w_{i,j}$, which is usually a real number. The weights take on different interpretations depending on what the graph represents for example transportation costs, distance covered, frequency of information flow and many others [86]. Let $G = (V, E)$ be a graph, then a graph $G' = (V', E')$ is a subgraph of G if and only if $V' \subseteq V$ and $E' \subseteq E$.

Remark 2.1.3. In this study, we will be working with simple connected undirected graphs or networks unless stated otherwise.



(a) (b) (c) (d)

Figure 2.1: Classifications of graphs: (a) A simple graph. (b) A graph with self-loops and multiple edges. (c) A directed graph. (d) A weighted graph.

Definition 2.1.4 (Incidence, vertex adjacency and neighborhood)**.** Given a graph $G = (V, E)$, node v and edge e are incident if v is one of and e_2 are said to be incident if they both have vertex $v \in V$ as an end-points [86]. Two vertices v_i and v_j are adjacent if there exists an edge, e, connecting the two vertices, that is, $e = \{v_i, v_j\}$. With the understanding of adjacency, we can represent a network using a matrix known as adjacency matrix $\boldsymbol{A}$ [86]. More so, the neighborhood of a vertex $v \in V$ is a set of all vertices that are adjacent to v [86]. Mathematically, $N_G(v) = \{u \in V | uv \in E\}$.

Definition 2.1.5 (Degree of a vertex (k_v))**.** The degree of a vertex v is defined as the number of edges incident to it. A self-edge is counted as two edges. The degree of a node v is the number of nearest neighbors of v, that is, $k_v = |N_G(v)|$. If $k_v = 0$, then node v is said to be isolated in G, and if $k_v = 1$, then v is a leaf of the graph. The minimum degree $k_{min}(G) = min\{k_v | v \in G\}$ and the maximum degree $k_{max}(G) = max\{k_v | v \in G\}$. For a directed network, we consider two types of degrees, namely in-degree (k_v^{in}) and the out-degree (k_v^{out}), which are the number of edges pointing towards or departing from vertex v respectively [37]. The total degree k_v is $k_v = k_v^{in} + k_v^{out}$.

Definition 2.1.6 (Walks, trails and paths)**.** A walk in a graph is a sequence of edges (not necessarily distinct)

$$(u_1, v_1), (u_2, v_2), \ldots, (u_k, v_k), \quad \text{for which } v_i = u_{i+1} \ (i = 1, 2, \ldots, l-1). \tag{2.1.1}$$

A trail is a walk in which all the edges are distinct [42]. A walk of length k is referred to as a k-walk. We can compute the number of k-walks between any pair of nodes in a network using the entries of $\boldsymbol{A}^k$ where $\boldsymbol{A}$ is the adjacency matrix of a graph. On the other hand, a path of length l is a walk of the same length in which all the nodes and edges are distinct. A closed path is called a cycle [37]. In a connected graph, there exists one or more paths connecting v_i to v_j and the ones with minimum length are referred as shortest paths (also referred to as geodesic paths). The length of a shortest path is known as the shortest path distance. Let $p_l(v_i, v_j)$ denote a shortest path between vertices v_i and v_j, an irreducible set of shortest paths of length l is the set $P_l = \{P_l(v_i, v_j), P_l(v_i, v_r), \ldots, P_l(v_s, v_t)\}$ in which the endpoints of every shortest path $P_l(v_i, v_j)$ in the set are different. Each path in this set is referred to as an irreducible shortest path [38].

Definition 2.1.7 (Random walk)**.** A random walk (also known as drunkard's walk) is a stochastic process that describes a path that consists of a succession of random steps on some mathematical space such as graphs. An example of a simple random walk on a graph can be described in the following way: Suppose a walker starts off a given node i and then moves to one of the neighbouring nodes of i, say j, with a uniform probability. Once at j, the walker again moves to the next node following a similar process until the stopping node. The sequence of these randomly visited nodes is referred to as the random walk [86]. For a simple undirected graph, the uniform probability, $p_{i,j}$, with which a random walker moves from node i to j is given by

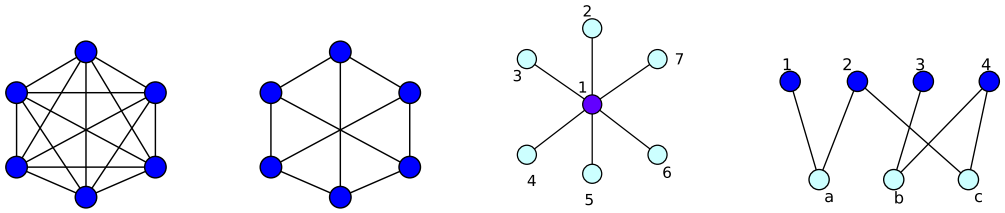
$$\boldsymbol{p}_{ij} = \frac{1}{k_i}\delta_{ij}, \tag{2.1.2}$$

where k_i denotes the number of neighbours of i and δ_{ij} is Kronecker delta which is equal to 1 if nodes i and j are connected and 0 otherwise.

Definition 2.1.8 (Connectivity of a graph)**.** A non-empty graph G is said to be connected if there exists a path between any two pair of vertices [86]. A connected component of an undirected graph is a subgraph in which any two vertices are connected to each other by paths, and which is connected to no additional vertices in the supergraph [86].

2.1.9 Categories of graphs

Here, we discuss some of the categories of graphs: A tree is a connected undirected graph that contains no closed loops [86]. It is important to note that a tree with n vertices has $(n-1)$ edges. A spanning tree of a graph $G = (V, E)$ is a subgraph of G with vertex set V, which is a tree. G has a spanning tree if and only if G is connected. A star graph of order n, denoted as S_n, is a tree on n nodes with one node having vertex degree $n-1$ and the other $n-1$ having vertex degree 1 [117]. A complete graph $G = K_{V(G)}$, on the other hand, is a graph on $V(G)$, if every two nodes are adjacent: $E = E(G)$.
A complete graph of order n is denoted by K_n. Moreover, a graph G in which every node has the same degree is referred to as a regular graph. A k-regular graph is one in which every node has degree equal to k. A cycle graph is a connected graph in which there exists an edge connecting one node to another and each node has degree 2. A cycle with n nodes is denoted as C_n [117]. Another interesting graph is a bipartite whose nodes can be divided into two disjoint sets V_1 and V_2 such that $(u, v) \in E$ implies that $u \in V_i$, $v \in V_j$, $i \neq j$. A bipartite graph in which each node of V_1 is connected to each node of V_2 is known as a complete bipartite graph; if $|V_1| = m$ and $|V_2| = n$, such a graph is denoted by $K_{m,n}$, for example, a star graph S_n is a complete bipartite graph $K_{(1,n-1)}$ [86].



(a) (b) (c) (d)

Figure 2.2: Categories of graphs: A complete graph with 6 nodes, denoted as C_6 in (a). A 3-regular graph (b). A star graph, S_6 in (c). A bipartite graph with $m = 4$ and $n = 3$ (d).

Definition 2.1.10 (Network)**.** Let us begin by considering a finite set $V = \{v_1, v_2, \ldots, v_n\}$ of unspecified elements, and let $V \otimes V$ be the set of all ordered pairs $[v_i, v_j]$ of the elements of V. A relation on the set V is any subset $E \subseteq V \otimes V$. The relation E is symmetric if $[v_i, v_j] \in E$ implies $[v_j, v_i] \in E$, and it is reflexive if $\forall v \in V, [v, v] \in E$. The relation E is antireflexive if $[v_i, v_j] \in E$ implies $v_i \neq v_j$. A network is a triple $G = (V, E, f)$, where V is a finite set of nodes, $E \subseteq V \otimes V = \{e_1, e_2, \ldots, e_m\}$ is a set of links, and f is a mapping which associates some elements of E to a pair of elements of V, for instance, if $v_i, v_j \in V$, then $f : e_1 \to [v_i, v_j]$ and $f : e_2 \to [v_j, v_i]$. For an undirected network, $[v_i, v_j] \in E$ implies $[v_j, v_i] \in E$ [37].

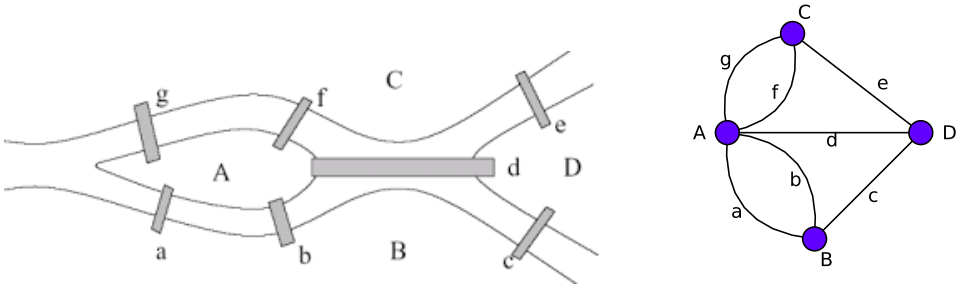
2.2 Complex systems and complex networks

Complex systems are very vital in our daily lives. They exist in fields such as social, economic, science, technology among others. During his interview with San Jose Mercury News on January 23, 2000, Stephen Hawking referred to the 21st century as one of complexity [59].

Complex systems are composed of interconnected components, however, it has been observed that many complex systems display a behaviour phenomenon (also known as emergent behaviour) that cannot be explained by any conventional analysis of the system's constituent parts [22]. In other words, for one to understand the behaviour of a system, it is necessary for one to consider a holistic system-level view point. There are different approaches to study of complex systems such as statistical description, empirical data analysis, simulations, analytical and network approach.

In this work, we focus on the network approach in which complex networks are used as diagrammatic representations of complex systems where nodes (vertices) represent system components or entities while links (edges) represent interconnections among the entities respectively. For example, a transportation system can be represented by network where nodes are cities or towns and the links are roads, railways or flight routes. This is then followed by mathematical formulation of the problem, modelling and validation.

An interesting early historical application of the network approach to the study of a system is the Königsberg bridge problem which is described in [43, 44]. Euler solved the problem by reformulating it in terms of a graph where vertices represent islands while edges represent the seven bridges joining any two islands as shown in Figure 2.3. Work published by Leonhard Euler [43] is considered the genesis of the story of network theory.



(a) (b)

Figure 2.3: The Königsberg bridges: (a) is a schematic diagram of the seven Königsberg bridges. (b) is a graph representing the Königsberg bridges. Source: [15].

2.2.1 Examples of real-world networks

Networks are used in many fields such as in biology, chemistry, computer science, transport, psychology, social sciences among others. For instance, in computer science, a network can be a representation of computers, routers, or any other electronic devices that are connected together by wires or wireless connections. In his work [87], Newman considered a loose categorisation of networks: social, communication, technological, and biological.

a) Social networks

   Networks considered as social networks are ones whose nodes correspond to people or groups of people while the edges represent the interactions or relationship between them [62]. For instance friendship networks such as facebook in which the interactions represent friendship ties among acquaintances, networks of intermarriages between families, social interaction networks which capture peoples' interactions through social activities or events, employee networks with companies, and many others. Some common networks that researchers have frequently experimented upon include: the Zachary karate network [51] which consists of two communities centred at the administrator and instructor as a result of misunderstandings that broke out among the members of the club. The nodes in the network are the members of the club as the links represent interactions between members during non-club activities. Other networks include terrorist networks [80] and many others.

b) Information networks:

   Information networks are also referred to as knowledge networks. Examples of networks under this category include: The world wide web which consists of billions of

web pages as nodes that are linked together through links known as hyperlinks [61], citation networks that are composed of nodes which are articles while directed link between two nodes denoted by $i \longrightarrow j$ indicate article i cites article j.

c) Technological networks:

This category consist of networks made by man to aid in distribution or transfer of resources, services or commodities such as electricity, water, transportation services, and many others. Examples of such networks include the internet, transportation networks, power grids, to mention but a few [9, 45, 92].

d) Biological networks:

Biological networks exists in areas related to human and processes that take place with in the human body, animals and their ways of survival, chemistry. Such networks include the human brain network, protein-protein interaction network, network of metabolic path ways, ecological networks [37, 95, 101].

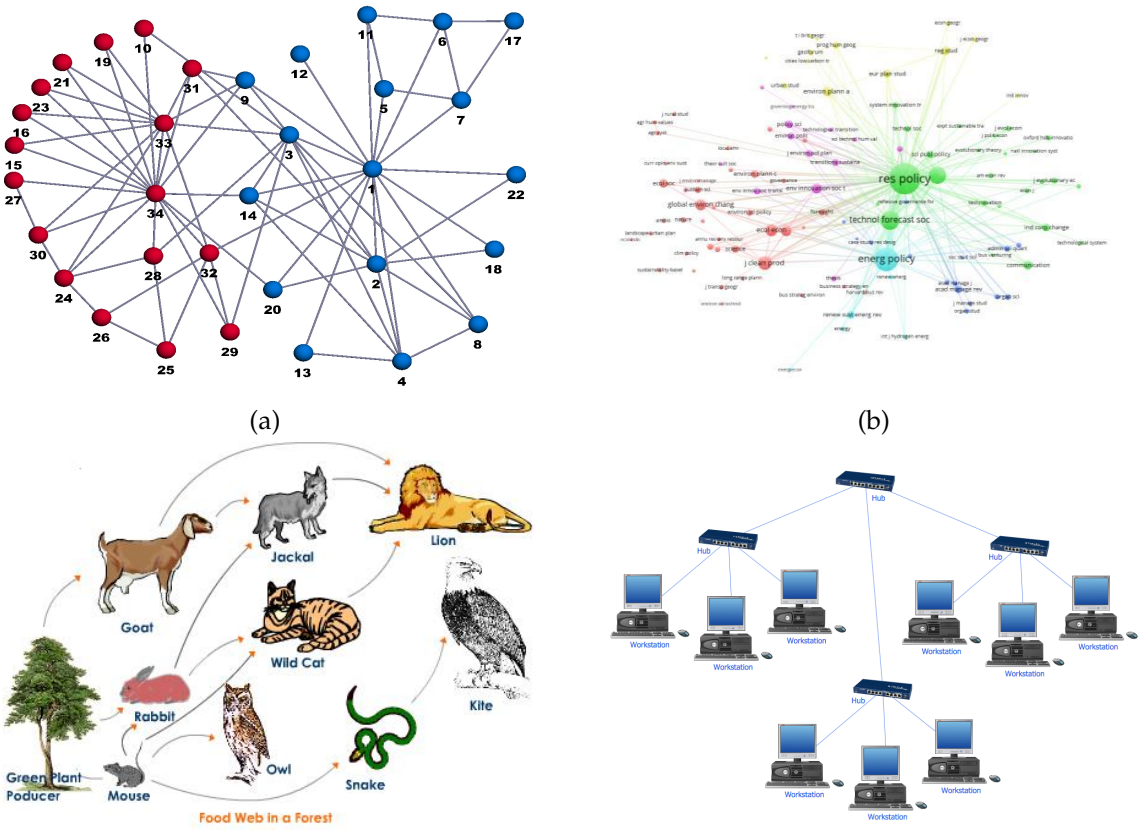


Figure 2.4: Sample networks in real-world: (a) A social network. (b) A co-citation network of journals. (c) A food web. (d) Computer network. Sources: [5, 51, 90, 110]

2.3 Matrix representations of graphs

Graphs can be represented in a numerous ways which include edge lists, matrices, and many others. However, matrices are the most widely used technique for representation of graphs especially for large graphs whose structure cannot be captured by human eye. In addition, representing graphs by matrices enables the application of mathematical and computer tools on networks for purposes of summation, pattern identification and many others [108]. In the subsequent subsections, we discuss the most common matrices used in the field of graph theory and their properties.

2.3.1 Adjacency matrix

The adjacency matrix (also known as binary adjacency) is a simple but very useful matrix commonly used in graph representation. It captures the connection between nodes in the graph, that is, the edge structure of a graph. For a graph with n nodes, the adjacency matrix $\boldsymbol{A}$ is an $n \times n$ matrix with elements

$$\boldsymbol{A}_{ij} = \begin{cases} 1 & \text{if } i \text{ and } j \text{ are adjacent,} \\ 0 & \text{otherwise.} \end{cases} \tag{2.3.1}$$

The summation of the i-th row or column is equivalent to the total number of immediate neighbours, known as degree of vertex v_i. For simple undirected networks, the matrix is symmetric with zeros entries at the main diagonal. However, for directed networks, the matrix may be asymmetric since edge directions have to be considered. For multi-graphs, the entries of $\boldsymbol{A}$ are the number of edges between each pair of vertices and for graphs with self-loops the diagonal entries are non-zero as self-loops may be counted once or twice based on whether the network is directed or undirected [16, 49]. For graphs with weighted edges, the adjacency matrix is given by

$$\boldsymbol{A}_{ij} = \begin{cases} w_{i,j} & \text{if } i \text{ and } j \text{ are adjacent,} \\ 0 & \text{otherwise,} \end{cases} \tag{2.3.2}$$

where $w_{i,j}$ denotes the weight of edge connecting nodes i and j.
The spectrum, which is the eigenvalues and their multiplicities, of the adjacency matrix is such a rich one and is thus used to mine interesting information about the graph. For example, the multiplicity of the largest eigenvalues is equal to the number of connected components of the graph [29].

The summation of the absolute values of the eigenvalues of the adjacency matrix is referred to as the energy, E, of the graph. It is given by $E = \sum_i |\lambda_i|$. The concept of energy of a graph has its origin in chemistry [52, 53] where it is used in approximating the total π electron energy of molecules [52]. This concept has however been extended to other matrices resulting into different kinds of graph energy as discussed in [52, 54]

2.3.2 Degree matrix

The degree matrix is a diagonal matrix that provides information about the degree of each node in a given network [86]. Given a network $G = (V, E)$ with $n = |V|$, the degree matrix $\boldsymbol{D}(G)$ is defined as

$$D_{i,j} = \begin{cases} k_i & \text{if } i = j \\ 0 & \text{otherwise,} \end{cases} \tag{2.3.3}$$

where k_i denotes the degree of node i. In a directed network the degree of node may be the in-degree or the out-degree or the sum of the two.

2.3.3 Distance matrix

The distance matrix also known as the all-pairs shortest path matrix denoted by $\boldsymbol{S}$ is a symmetric matrix whose elements are defined as

$$S_{i,j} = \begin{cases} l_{i,j} & \text{if } i \neq j \\ 0 & \text{otherwise,} \end{cases} \tag{2.3.4}$$

where $l_{i,j}$ is the length of the irreducible shortest path between nodes i and j.

2.3.4 Incidence matrix

Consider a graph with vertex set $V = \{v_1, v_2, \ldots, v_n\}$ and edge set $E = \{e_1, e_2, \ldots, e_m\}$. Let us consider an arbitrary orientation of every edge in the graph, say, we label each edge $\{v_i, v_j\}$ in a way that v_i is the positive end and v_j is the negative end. It should be, however, noted that the orientation does not matter. Then the oriented incidence matrix $\boldsymbol{B}(G)$ has entries defined as

$$B_{ij} = \begin{cases} +1 & \text{if node } v_i \text{ is the positive end of the edge } e_j \\ -1 & \text{if node } v_i \text{ is the negative end of the edge } e_j \\ 0 & \text{otherwise.} \end{cases} \tag{2.3.5}$$

2.3.5 Laplacian Matrix

The Laplacian matrix or graph Laplacian is another matrix used in representation of graphs. Recently, a number of researchers have been deeply involved in the study of the Laplacian matrix of a graph since this matrix has interesting spectral properties that provide more useful information about the structure of a graph as compared to other matrices such as the adjacency matrix. The Laplacian plays a key role as a natural link between discrete representations like graphs and continuous representations such as vector spaces and manifolds. There are various applications of the graph Laplacian which include spectral clustering, spectral matching, diffusion on networks, centrality measure, among others. We will discuss the diffusion process on networks and the role of the Laplacian matrix of a graph in chapter 3.

The Laplacian matrix takes on different versions namely the normalised and unnormalised Laplacian matrices. We thus discuss some of the variations of the Laplacian matrix:

Definition 2.3.6 (Combinatorial Laplacian Matrix)**.** Given a graph $G = (V, E)$, its combinatorial Laplacian or unnormalised Laplacian matrix is defined as the difference between the degree matrix $\boldsymbol{D}$ and the adjacency matrix $\boldsymbol{A}$ of the graph. That is, $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$. The entries of $\boldsymbol{L}$ are given by

$$L_{ij} = \begin{cases} k_{v_i} & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } v_i \text{ is adjacent to } v_j \\ 0 & \text{otherwise,} \end{cases} \tag{2.3.6}$$

where k_{v_i} denotes the degree of node i [37]. For a directed graph, the Laplacian matrix can be defined in different ways for instance diagonal entries may be the out degrees, in degrees, the sum or average of both, depending on the application.

Alternatively, the combinatorial Laplacian matrix can be defined in terms of the vertex-edge incidence matrix $\boldsymbol{B}$. That is,

$$\boldsymbol{L} = \boldsymbol{B}\boldsymbol{B}^T, \tag{2.3.7}$$

where $\boldsymbol{B}^T$ is the transpose of $\boldsymbol{B}$ [37].

In most literature, the combinatorial Laplacian matrix is referred to as the Laplacian matrix. We will use the same convention in this thesis. The combinatorial Laplacian matrix is a real, symmetric and singular matrix. It is also positive semi-definite, that is to say, all its eigenvalues are non-negative. This property makes the Laplacian matrix more suitable for spectral analysis compared to the adjacency and incidence matrices [30].

2.3.6.1 Spectrum of the combinatorial Laplacian matrix

The spectrum of the Laplacian matrix is the set of all its eigenvalues and their multiplicities [37]. Let $\lambda_1 < \lambda_2 < \ldots < \lambda_n$ be the distinct eigenvalues of $\boldsymbol{L}$ and let $m(\lambda_1), m(\lambda_2), \ldots, m(\lambda_n)$ be their multiplicities. Then, the spectrum of $\boldsymbol{L}$ is written as

$$Sp(\boldsymbol{L}) = \begin{pmatrix} \lambda_1 & \lambda_2 & \ldots & \lambda_n \\ m(\lambda_1) & m(\lambda_2) & \ldots & m(\lambda_n) \end{pmatrix}. \tag{2.3.8}$$

For some categories of simple graphs, the following expressions of their spectra have been put forward [37]:

- Star, S_n : $Sp(\boldsymbol{L}) = \{0,\ 1^{n-2},\ n\}$.
- Complete, K_n : $Sp(\boldsymbol{L}) = \{0,\ n^{n-1}\}$.
- Complete bipartite, $K_{m,n}$: $Sp(\boldsymbol{L}) = \{0,\ m^{n-1},\ n^{m-1}\}$.

We note that the summation of eigenvalues, that is to say, the trace of $\boldsymbol{L}$ is twice the number of edges, E, of a graph [19]. Mathematically, $\sum_{i=1}^{n} \lambda_i = \text{Trace}(\boldsymbol{L}) = 2|E|$. Given a graph, G, with n connected components denoted by G_i for $1 \leq i \leq n$, its spectrum is the union of the spectra of the components G_i with multiplicities inclusive [19].

As mentioned earlier, the spectrum of the Laplacian matrix provides useful information pertaining the structure of a network. Firstly, the smallest eigenvalue of the Laplacian matrix is zero and its multiplicity is equal to the number of connected components of the corresponding graph. A graph is connected if its second smallest eigenvalue is nonzero. That is, $\lambda_2 > 0$ if and only if G is connected. The eigenvalue λ_2 is thus called the algebraic connectivity of a network, $a(G)$ whose magnitude tells how well connected a graph is. The algebraic connectivity has significant implications for properties such as clustering and synchronizability. The eigenvector corresponding to the eigenvalue λ_2 is called the Fiedler vector [42]. By Kirchoff's matrix-tree theorem, the number of unique spanning trees of a connected graph is equal to the value of any cofactor of the matrix $\boldsymbol{L}$ [58].

2.3.7 Spectral matrix of the Laplacian

The spectral matrix results from the eigenvector expansion of the Laplacian matrix $\boldsymbol{L}$ such that its columns are the scaled eigenvectors, that is

$$\boldsymbol{\Phi} = (\sqrt{\lambda_1}\boldsymbol{v}_1, \sqrt{\lambda_2}\boldsymbol{v}_2, \ldots, \sqrt{\lambda_n}\boldsymbol{v}_n), \tag{2.3.9}$$

where λ_i and v_i are the i-th eigenvalues and corresponding eigenvectors respectively [116]. The matrix $\boldsymbol{\Phi}$ is quite interesting as it is a complete representation of the graph. Moreover, we can reconstruct the original Laplacian matrix from $\boldsymbol{\Phi}$ using the formula $\boldsymbol{L} = \boldsymbol{\Phi}\boldsymbol{\Phi}^T$. It is however important to note that the matrix $\boldsymbol{\Phi}$ is a unique representation of $\boldsymbol{L}$ iff all n eigenvalues are distinct or zero which implies that the eigenvectors are distinct as well. Wilson and Hancock [116] discuss in detail the use of spectral matrix in graph clustering. We explore the concept of graph clustering in chapter 4 .

2.3.8 Normalized Laplacian matrix

One format of normalized Laplacian matrix also known as symmetric normalised Laplacian is defined as

$$\boldsymbol{\mathcal{L}} = \begin{cases} 1, & \text{if } i = j \text{ and } k_i \neq 0, \\ -\frac{1}{\sqrt{k_i k_j}}, & \text{if } i \text{ and } j \text{ are adjacent,} \\ 0, & \text{otherwise.} \end{cases} \tag{2.3.10}$$

We can also write

$$\boldsymbol{\mathcal{L}} = \boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2} = \boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2} \tag{2.3.11}$$

where $\boldsymbol{D}^{-1/2}$ is the diagonal matrix whose diagonal entries are the inverse square root of he respective diagonal entries of the degree matrix [37]. For a k-regular graph, we have

$$\boldsymbol{\mathcal{L}} = \boldsymbol{I} - \frac{1}{k}\boldsymbol{A}. \tag{2.3.12}$$

The matrix $\boldsymbol{\mathcal{L}}$ is symmetric with real and non-negative eigenvalues which satisfies:

$$0 = \lambda_1(\boldsymbol{\mathcal{L}}) \leq \lambda_2(\boldsymbol{\mathcal{L}}) \leq \ldots \leq \lambda(\boldsymbol{\mathcal{L}}) \leq 2.$$

It is very convenient to work with the spectrum of the normalised Laplacian because of the small interval $[0, 2]$ for instance, it is easier to compare the spectrum of the normalised than the combinatorial Laplacian of two graphs. In addition, the eigenvalues in some areas such as stochastic process are consistent with the spectrum of the normalised Laplacian matrix. More over, when the spectrum of the normalised Laplacian is used, it is possible to generalise results that were only known for regular graphs to other categories of graphs [26]. It is for these reasons that we will, in most cases, consider the normalised version of the Laplacian.
We note that the largest eigenvalue, $\lambda_n(\boldsymbol{\mathcal{L}})$, is equal to 2 only for a bipartite graph. Like the combinatorial Laplacian, the smallest eigenvalue is zero and its the multiplicity is equal to the number of connected components of the corresponding graph.

2.3.9 Random walk normalised Laplacian

This another variant of the Laplacian matrix that is based on random walks in the graph. It is defined as

$$\boldsymbol{\mathcal{L}}_r = \boldsymbol{L}\boldsymbol{D}^{-1} \tag{2.3.13}$$

The relationship between $\boldsymbol{\mathcal{L}}_r$ and $\boldsymbol{\mathcal{L}}$ is :

$$\boldsymbol{\mathcal{L}}_r = \boldsymbol{D}^{1/2}\boldsymbol{\mathcal{L}}\boldsymbol{D}^{-1/2}. \tag{2.3.14}$$

By diagonalizing $\boldsymbol{\mathcal{L}}$, equation 2.3.14 can be rewritten as

$$\boldsymbol{\mathcal{L}}_r = (\boldsymbol{D}^{1/2}\boldsymbol{\Phi})\boldsymbol{\Lambda}(\boldsymbol{D}^{1/2}\boldsymbol{\Phi})^{-1}, \tag{2.3.15}$$

where $\boldsymbol{\Lambda}$ is a diagonal matrix of eigenvalues of $\boldsymbol{\mathcal{L}}$ and $\boldsymbol{\Phi}$ is the eigenvector matrix. Thus, $\boldsymbol{\mathcal{L}}_r$ is diagonalizable and it shares the same set of eigenvalue as $\boldsymbol{\mathcal{L}}$ though the corresponding eigenvectors are different. Based on this relationship, it is possible to use $\boldsymbol{L}_r$ in place of $\boldsymbol{\mathcal{L}}$ in some applications. A detailed account of other variants of the Laplacian matrix can be found in [107].

2.4 Structure of a network

In the process of understanding the functionality of systems, studying the structure of the system is a fundamental step and this posses a question of what role the structure of a system plays in revealing insights about the its functionality. The answer to this question clearly

follows from a quote from Francis H. C. Crick who is the co-discoverer of the structure of the DNA molecule. It states that "If you want to understand function, study structure". We must know that it is not guaranteed that having known the structure then we definitely know how the system works, however, it is certainly impossible for one to understand the functionality of the system without prior knowledge of its structure.

The structure or topology of a network is a description of how nodes are connected to each other in a network. For example in a complete graph, each node is connected to every other node in the network, the star graph consists of a 'central' node to each other nodes are connected and there is link joining any pair of non-central nodes. The structure of a network is captured by the adjacency matrix (see equation 2.3.1). The study of network structure for purposes of understanding and predicting properties of networks was first and much embraced by chemists who used graphs to represent chemical molecules where vertices are used to represent atoms and edges to represent chemical bonds. In their work [20], Crum Brown and Fraser were among the earliest researchers who put forward the idea that the properties of a chemical molecule are greatly dependent on its structure. This structure-property relationship has further been explored using graph-theoretical approach not only in chemistry but in other fields such as social networks, biological networks and many others [114, 115].

For small networks made of tens or thousands of nodes, it is possible to ascertain the structure of such networks by analysis using direct use of human eye. However, as networks grow in size and complexity, the analysis of their structure require advanced analytic approaches. A number of statistical measures have been developed for purposes of characterising large networks. These quantifiers range from global measures which consider the whole graph, intermediate measures for groups of nodes and local measures which are applied to individual nodes. In the subsequent subsections, we discuss some statistical characterisation of the structure of network:

2.4.1 Diameter

The diameter of a graph, denoted as D, is the maximum distance between pairs of nodes in the graph[86]. The diameter of a graph is given by

$$D = \max_{i,j \in V} \{d_{ij}\}, \tag{2.4.1}$$

where d_{ij} is the shortest path between node i and j.

For a disconnected network, the diameter is undefined and therefore, for such a case, we take the efficiency which is defined as

$$\bar{e} = \frac{1}{n(n-1)} \sum_{i,j \in V, i \neq j} \frac{1}{d_{ij}}, \tag{2.4.2}$$

where $d_{i,j}$ is the shortest path between vertices i and j, and n is the size of the network.

The efficiency of a network is a measure of the efficiency of information flow between vertices, thus networks with high values of efficiency are considered fast spreaders of information and a large percentage of vertices are reached compared to their counterparts with lower values of efficiency.

2.4.2 Characteristic path length

The characteristic or average path length of a network is the average number of steps along the shortest paths for all possible pairs of network nodes. Given a network $G = (V, E)$, the average path length l_G is defined by

$$l_G = \frac{1}{n(n-1)} \sum_{i,j \in V,\, i \neq j} d_{ij}, \tag{2.4.3}$$

where d_{ij} is the shortest path between node i and j and n is the total number of nodes in G. The value of l_G determines the size of a network as well as the efficiency of information flow or disease spread over a network [86].

2.4.3 Clustering coefficient

The clustering coefficient of a network measures the degree to which nodes tend to cluster together. Such behaviour is more evident in real-world networks, especially social networks where nodes cluster together forming tightly knit groups that have relatively high density of ties among them [42]. Consider three nodes in a network labelled i, j and k. Suppose i is connected to both j and k, then the likelihood that j and k are also connected is known as the clustering coefficient. In other words, clustering coefficient measures the density of triangles in a network. The value of clustering coefficient lies in the interval $[0, 1]$. There are two types of clustering coefficients namely, the local and the global clustering coefficients.

Definition 2.4.4 (Local clustering coefficient)**.** The local clustering coefficient is a measure of the clustering tendency in a node's immediate neighborhood. The local clustering coefficient for a node i with degree k_i is formally defined as

$$C_i = \frac{\text{number of pairs of neighbors of } i \text{ that are connected}}{\text{number of pairs of neighbors of } i} = \frac{2t_i}{k_i(k_i - 1)}, \tag{2.4.4}$$

where t_i denotes the number of triangles attached to node i. For nodes with degree equal to zero or one, we set $C_i = 0$ since there are no triangles attached to such nodes [86]. The average clustering coefficient of a network is given by

$$\bar{C} = \frac{1}{n} \sum_i C_i. \tag{2.4.5}$$

Definition 2.4.5 (Global clustering coefficient)**.** The global clustering coefficient measures the density of triplets of nodes in a network where a triplet is composed of three nodes that are

connected by either two (open triplet) or three (closed triplet) ties or links. The global clustering coefficient determines the overall level of clustering in a network [86]. Mathematically, we define the global clustering coefficient,C, as

$$C = \frac{3 \times \text{number of triangles}}{\text{number of connected triplets of vertices}} = \frac{\sum t_{\triangle}}{\sum t}, \tag{2.4.6}$$

where $\sum t_{\triangle}$ is the total number of closed triplets and $\sum t$ is the total number of connected triplets of nodes in a network.

2.4.6 Degree distribution of networks

The scattering of node degrees over a network is characterised by the distribution function, $p(k)$, which is the probability that a node chosen uniformly at random has degree, k. We define $p(k)$ to be the fraction of nodes in a network that have degree k. That is, $p(k) = n(k)/n$, where $n(k)$ is the number of nodes with degree k in a network of size n. The degree distribution of a network is referred to as the probability distribution of node degrees over that network. It is represented by plot of $p(k)$ against k [37]. Figure 2.5 is of plots of some of the common degree distributions in networks namely Gaussian, Poisson, exponential and power-law degree distributions.

2.4.6.1 Power-law degree distribution

In the power-law degree distribution, there exists few nodes with very high degree and many with very low degree. In other words, the probability of finding a node with degree k decreases as a negative power of degree k. This implies that in such networks, it is less likely to find a node with high degree [37]. Formally,

$$p(k) = Ck^{-\gamma}, \text{ for } 2 \leq \gamma \leq 3. \tag{2.4.7}$$

The range of the exponent between 2 and 3 was asserted by Barabási in Linked [10]. However, other literature indicate an exponent of value greater than zero (that is, $\gamma > 0$).
Using a logarithmic scale, the plot of equation 2.4.7 is a straight line,with a slope equal to $-\gamma$ and an intercept equal to $\ln C$ and it is given by $\ln p(k) = -\gamma \ln k + \ln C$.
However, we observe that the part that corresponds to high degrees (tail of the distribution) is very noisy. One solution to overcome the problem is to consider the cumulative distribution function, which is defined as

$$P(k) = \sum_{k'=k}^{\infty} p(k'),$$

which represents the probability of randomly choosing a node with degree k or greater [37]. When we scale the degree by a constant factor a, we obtain

$$p(k, a) = C(ak)^{-\gamma} = a^{-\gamma} p(k) \propto p(k). \tag{2.4.8}$$

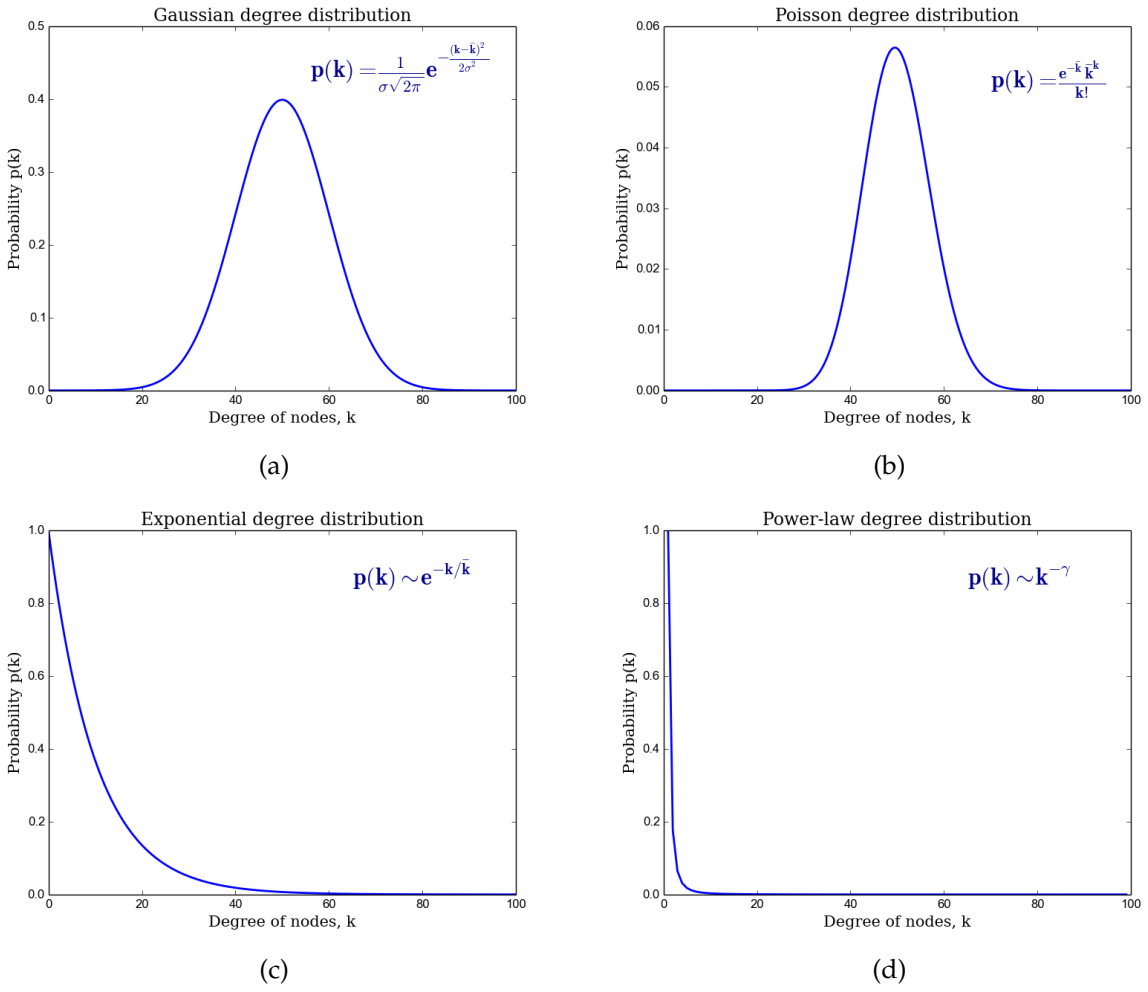


Figure 2.5: Common degree distributions of networks: (a) Gaussian distribution. (b) Poisson distribution. (c) Exponential distribution.(d) Power-law distribution.

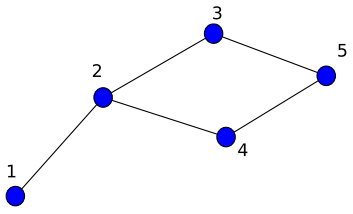
Thus, scaling by a constant a multiplies the original power-law relation by the constant $a^{-\gamma}$ which implies that all power laws with a particular scaling factor are scaled versions of each other and thus networks that follow a power-law distribution are referred to as scale-free networks.

In network science, there is an ambiguity in language regarding what 'scale-free' really means. Some possible suggestions of what a scale free network is include: a network with power degree distribution with exponent in range 2 to 3, a network formed by the process of preferential attachment where nodes joining the network connected to already existing high degree nodes in the network, one that follows a power law on some or all scales, and many others. Work published by Barabási and his then graduate student Réka Albert [9] claimed that many real-world networks (such as genetic networks and world-wide web) have node connectivities that follow a scale-free power-law degree distribution. Since then, a multitude of papers have been published affirming the existence of scale-freeness in biological, social, and many other real-world networks [5, 27, 64].

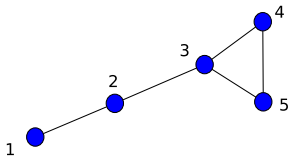
However, the claim that scale free networks are universal has been criticised by some network scientists and experts. For example, the scale freeness of some networks such as the power networks and metabolic networks has been questioned [6, 103]. The most recent criticism was raised by Clauset and Broido in [18] where they asserted that scale free networks are rare in nature based on statistical analysis of nearly 1,000 networks from social, technological, biological and many other domains. A very interesting debate on this issue was captured in [70] which indicates that there is need for research directed towards that area to clear the ambiguities regarding scale-freeness and the universality of scale-free networks.

2.4.6.2 Structure of network and its degree distribution

The degree distribution of a graph provides useful insights about its structure though it is not conclusive as graphs with different structures can have the same degree distribution which implies that degree distribution gives us some but not all the information regarding the structure of a graph. Thus, in most cases, we cannot deduce a complete structure of a network based on knowledge of its degree distribution. To further backup this argument, we consider a simple example of two graphs both of size 5 (Figure 2.6a & Figure 2.6b) and whose degree distribution is shown in table 2.6c.



(a)



(b)

Degree, k	$p(k)$
1	1/5
2	3/5
3	1/7

(c)

Figure 2.6: Two simple graphs (a and b) with the same degree distribution given in table (c).

We then compute some of the structural properties of the two graphs (as in table 2.1) to ascertain whether their structures are the same.

	Diameter	Average degree	Clustering coefficient	Global efficiency
G1	3	1.6	0.0000	0.7333
G2	3	1.7	0.4667	0.7167

Table 2.1: Summary of values of some of the structural properties of the two simple graphs (Figures. 2.6a & 2.6b) whose degree distribution is the same.

Table 2.1 contains some of the selected structural properties computed for the two graphs.

First, the diameter of both graphs is the same. However, other properties such as average degree, clustering coefficient and global efficiency have different values for both graphs. For example graph $G1$ has clustering coefficient of 0 yet graph $G2$ has a value of 0.4667. From the two values, we can tell that the structure for the two graphs is different. We can thus conclude that we cannot totally rely on the similarity in degree distribution for both graphs to deduce that the graphs under study are structurally similar since other structural properties indicate otherwise.

2.5 Random models of networks

In the study of real-world networks such as protein-protein interaction, transport networks, and food web, we observe different topological structures of these networks and it is of great importance to understand the mechanisms that are responsible for such structures. We therefore discuss some of the models that mimic real-world networks.

2.5.1 The Erdős-Rényi (ER) model

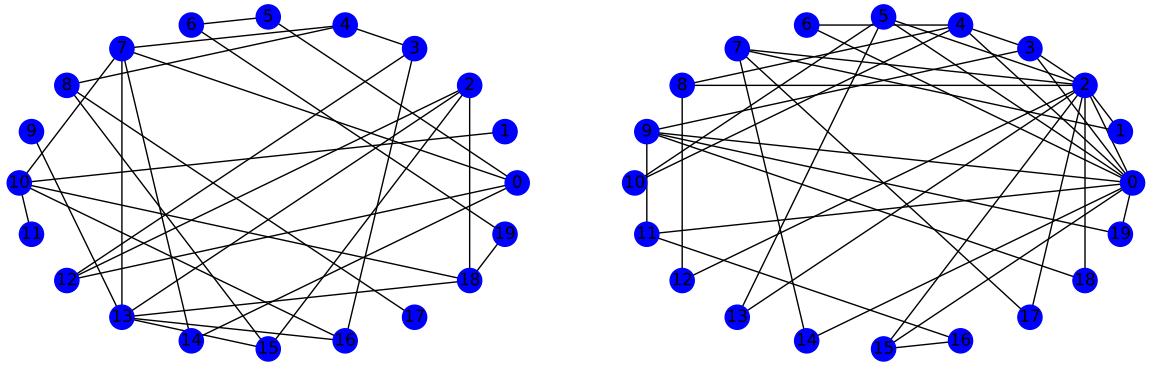
This random graph model was introduced by Erdős and Rényi in 1959. It is the best known model for random networks. In this model, we take n isolated nodes and then fix a probability p with which we link the nodes. For each pair of nodes, we generate a random number, r, uniformly from the interval $[0, 1]$. If $p > r$, the two nodes are connected forming a network. Networks generated by the ER model have a small average path length and their average clustering coefficient tends to zero as n increases [37].

2.5.2 The Barabási-Albert (BA) model

The motivation for developing this model was the claim that most of the real-world networks follow a power-law degree distribution which did not manifest in ER networks and thus, Barabási and Albert tackled this curiosity by developing a network model that generates networks which mimic the degree distribution observed in the real-world networks. This model uses a preferential attachment (also known as rich get richer) mechanism to evolve a given initial graph. The preferential attachment technique commences with an initial network with n_0 nodes. This is then followed by introducing a new node at each time step. The new node connects to the existing nodes with a probability proportional to the degrees of existing nodes. In other words, it is highly probable that the incoming node connects to the high degree existing nodes. The resulting networks have high clustering coefficients and are scale-free [37]. We note that the BA networks created using the above code follow a power law degree distribution $p(k) \sim k^{-3}$ [37].

In this study, we will perform simulations on networks developed by the mentioned models using the code snippet implemented in NETWORKX package in PYTHON [55]:

```
import networkx as nx

n = 20              #number of nodes of the graph
p = 0.15             #probability for edge creation
m = 30               #number of edges
m_b = 2              #number of edges connecting  a new node to existing nodes
seed = None          #seed for random number generator

#creation of the 3 undirected connected graphs namely  random graphs //
#using G_np, G_n,m and BA models and if resulting graph is //
#not connected, repeat process till a connected one is obtained

G_np = nx.gnp_random_graph(n,p,seed,directed=False)
while (nx.is_connected(G_np)==False):
        G_np = nx.gnp_random_graph(n,p,seed,directed=False)

G_nm = nx.gnm_random_graph(n, m, seed, directed=False)
while (nx.is_connected(G_nm)==False):
        G_nm = nx.gnm_random_graph(n, m, seed, directed=False)

G_ba = nx.barabasi_albert_graph(n,m_b,seed)
while (nx.is_connected(G_ba)==False):
        G_ba = nx.barabasi_albert_graph(n,m_b,seed)

#visualise the 3 graphs
nx.draw_circular(G_np, with_labels=True, node_color='b'); plt.show()
nx.draw_circular(G_nm, with_labels=True, node_color='b'); plt.show()
nx.draw_circular(G_ba, with_labels=True, node_color='b'); plt.show()
```



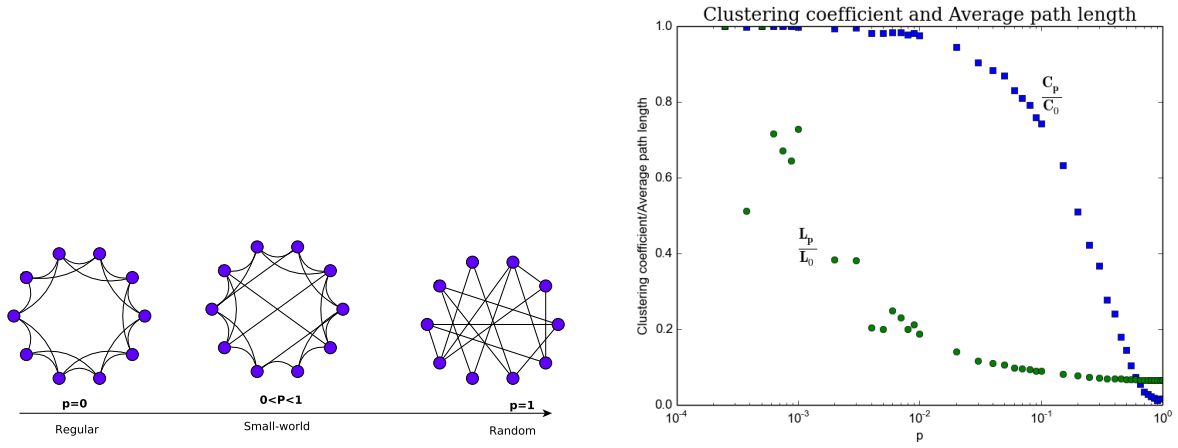
(a) (b)

Figure 2.7: Two sample graphs created using the ER and BA models implemented in networkX. (a) is an ER graph with $n = 20, p = 0.15$ and (b) is a BA graph with $n = 20, m_b = 2$.

We compare the structures of the two graphs generated by two different models. 2.7a is a ER graph with degree of nodes in the range $[1, 5]$ which implies that there is no much difference between degrees of nodes, that is to say homogeneous graph. On the other hand, the BA graph in 2.7b has most of the nodes with degrees in the range $[2, 5]$. However, we notice that a few nodes namely nodes 2 and 0 have very high degrees compared to the rest of the nodes. These two nodes are referred to as hubs.

2.5.3 The Watts-Strogatz (WS) model

An experiment carried out by Stanley Milgram in 1967 [83] consisted of randomly selected people in the cities of Omaha and Wichita in the United States of America (USA) who were asked to send letters to target persons living in Boston. The individuals at the starting points were asked to send the letters to persons they knew. Despite the fact that the senders and their respective targets were separated by more than 2000 kilometres and that there was a total of 200 million inhabitants in the USA by then, results showed that the letters took an average of about six steps to reach the targets and that there was a large group interconnection where an acquaintance of an individual fed back into his own circle, thus eliminating new contacts. In terms of networks, two properties are observed: the former implies a small average path length of 6 while the latter implies a high clustering coefficient. It is from this experiment that the phrase "six degrees of separation" was born. This phrase explains a small world in which an individual is connected to another by a maximum of six steps. In 1998, the authors in [112] proposed a model that reproduces the two properties mentioned previously. This model starts with a circulant (or ring) network with n nodes connected to k nearest neighbors. With a fixed probability p, an end to each original link is rewired to a new randomly selected node. The WS model interpolates between a regular and a random network as shown in Figure 2.8(a). The intermediate network is a small-world characterised by high clustering coefficient and small average path length.



(a) (b)

Figure 2.8: The rewiring process: (a) Interpolation of WS model as probability increases. (b) Illustration of the variation of clustering coefficient and average path length during the rewiring process.

In Figure 2.8(b), as the probability increases during the rewiring process, the average path length and clustering coefficient vary. The region where the average path length is small and the clustering coefficient is high corresponds to the small world network in Figure 2.8(a).

2.6 Graph similarity

Graph similarity is a way of comparing a given set of graphs in order to identify similarity between them. However, similarity can be defined in different ways depending on the application for example graphs can be considered similar if they are isomorphic, that is to say, they are identical copies of each other. In cases where isomorphism is impossible, similarity between graphs can be based on the cost incurred in changing one graph into another. These changes can be in form of edge or node deletion or addition, edge redirection in directed edges, among others. Alternatively, one may quantify graph similarity based on how the neighborhood of a given node in one graph is similar to the neighborhood of another node in the other graph [120]. With these and many other definitions, the concept of graph similarity remains ambiguous. Despite the ambiguity, graph similarity has been used in a multitude of applications such as comparison of biological networks, web searching, chemical structure matching, synonym extraction, image clustering, social network mapping and many others [89, 120]. We note that the similarity measure used is selected guided by the intended application. In literature [120], measures of similarity have been categorised as follows:

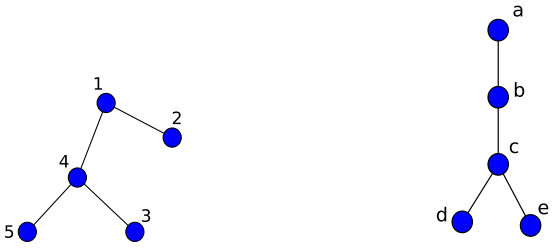
1. **Isomorphism based techniques**

   Let G and H be graphs with vertex sets $V(G)$ and $V(H)$ respectively, are isomorphic if there exists a bijective (one-to-one and onto) function, denoted by f, between the vertex

sets

$$f : V(G) \longrightarrow V(H), \tag{2.6.1}$$

such that any two vertices are connected in one graph if and only if their images under the bijection are connected [120]. This bijection must preserve the edges in graphs and thus the two are structurally identical as illustrate in Figure. 2.9. The case where the bijective function acts on nodes of the same graph is known as graph automorphism which is the mapping off a graph to itself.



(a) (b)

Figure 2.9: Two simple isomorphic graphs

Alternatively, similarity of graphs can be based on isomorphism of the subgraphs of corresponding graphs for instance two graphs are considered similar if they both contain common subgraph and the larger the subgraph, the higher the similarity between the graphs. Another case involves the measure of similarity based on whether one graph is isomorphic to a subgraph of another graph. This kind of isomorphism is known as subgraph isomorphism. In image analysis specifically scenery study, subgraph isomorphism has been used to ascertain whether a given object is part of another object or a group of objects. Detailed algorithms for the three similarity measures can be found in [75, 109, 113].

In cases where the above methods cannot be applied, an alternative is inexact matching which involves modification of one graph labelled source graph to obtain another graph called the target graph. One such method is graph edit-distance which involves transforming one graph into another by performing edit operations such as edge or node deletions, insertions or substitutions, edge splitting, edge contraction, among others. Each operation is associated with a cost and the sequence of operations with the minimum cost is attained which amounts to a measure of similarity between the two graphs [48]. Formally, a source graph G and target graph H, the graph edit distance is given by

$$d(G,H) = \min_{(e_1,e_2,\ldots,e_n)\in\tau(G,H)} \sum_{i=1}^{n} c(e_i), \tag{2.6.2}$$

where $\tau(G,H)$ is the set of edit operations that transform G into H, and $c(e_i) \geq 0$ is the cost of graph edit operation e_i.

For small number of vertices of a given graph, exact algorithms such the A* algorithm can be feasibly used to compute the edit distance. However, for large graphs, such algorithms are not practical due to high computation complexity and thus heuristic algorithms are instead considered as option. According to Kubicka et al., finding the global optimal of the edit distance problem is Non-deterministic Polynomial-complete [74].

2. **Iterative methods**

   These methods are based on the idea of node similarity for which two nodes are similar if their neighbourhoods are similar. The methods involve iterative processes in which nodes share similarity scores at each iteration until convergence is attained. There are various algorithms used in computing the similarity using the iteration method, however, the difference mainly lies in the rules used to update similarity scores. A detailed review of some of these algorithms can be found in [82, 120].

   We note that both graph isomorphism and iterative methods require a correspondence between nodes of two graphs to be established which, as we mentioned earlier on, is a cumbersome task especially for large graphs. Another draw back associated with these methods is scalability which implies that methods do not perform well on very large networks.

3. **Feature extraction**

   These are statistical methods that measure graph similarity based on properties of graph structure such as degree distribution, betweenness measures, diameter, graph spectrum and many others. Graphs are counted similar if their respective aggregated properties are assessed based on certain similarity measures and similarity is established. This method is very powerful as it maps graphs to various statistics that are smaller and easier to work with compared to original graphs. However, one draw-back of this method is the choice of the statistics which at times may not give intuitive results. Take an example of graphs with different number of nodes but with the same diameter, these graphs may be considered similar which is not actually true. In most of the literature, graph properties based on eigenvalues and eigenvectors of the adjacency and Laplacian matrices have been intensely explored. For example, in [78] a vector of k leading eigenvalues of the adjacency matrix of a graph is used to describe the graph, elements of the spectral matrix are used in [116], among others. In later chapters, we will discuss other graph similarity measure based on the Laplacian spectrum.

In this work, we use the concept of graph similarity as a basis for performing clustering (also known as graph clustering) on a given set of graphs extracted from objects. Here, we aim at grouping graphs into classes such that graphs belonging to the same class are similar and those belonging to different classes are dissimilar. To start with, we discuss some of the key concepts encountered in the extraction of graphs representing objects.

2.7 Corner point detection for an image

A feature can be defined as an 'interesting' part of an image based on the type of application or problem at hand. The detection of features of an image is a core step in a number of image related tasks such as object recognition, motion tracking, 3D scene reconstruction and many other tasks [106]. These feature points can be obtained from intensity, colour, texture, colour, among others. In this study, we are interested in intensity-based features. Taking a small window of pixels, a feature is categorised as a corner, edge or flat region based on the amount of change in pixel intensity on shifting the window in any direction as shown in Figure 2.10.

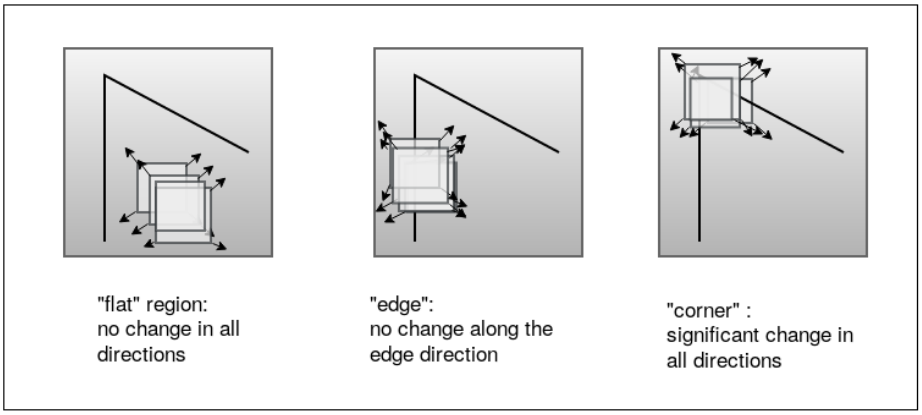


Figure 2.10: Illustration of the impact of shifting window to the pixels within that window using Harris corner detector.

From Figure 2.10, pixels can be categorised as either a flat region where all shifts of the window result in very small changes in intensity, an edge where a shift in the perpendicular results into a large change or a corner point where all shifts result into a large change in intensity and thus referred to as good feature points. According to [84], corner points are points where there is intensity variation in all directions. The corner points (or points of interest) are highly used in computer vision applications because for a particular image, thousands of these features can be obtained and real time performance be achieved using these corners. There are numerous corner point detection methods which include Moravec's corner detector [84], high-speed detector [94], Harris corner detector [57] and many others.

In this study, we are building on earlier work in [102, 118] where the Harris corner detector was used. Moreover, the Harris corner detector is simple to compute, fast and most importantly, the corner points obtained based on this method are invariant in position to rotation, scale, illumination, and partially to affine intensity changes as discussed in detail in [102]. one main drawback of the Harris corner detector is that its is not invariant to image scaling. However, solution to this problem has been discussed in [102].

2.7.1 The Harris corner detector

The Harris corner detector is a corner detection algorithm put forward by Stephen and Harris [57]. It is an improvement to the earliest algorithm introduced by Moravec [84]. According to Moravec [84], a corner is defined as a point with low self-similarity. Considering a window positioned at a given pixel and then subjecting to a slight shift in different directions, the similarity between the windows is obtained by computing the sum of squared differences (SSD) of corresponding pixels of both windows [84]. The pivotal pixel is thus categorised as a corner based on the similarity value obtained, that is to say, a lower value implies more similarity. The corner strength is the smallest SSD between the window and its neighbours in the horizontal, vertical and diagonal directions. One of the drawbacks of this algorithm as pointed out by Moravec [84], is that the algorithm is not isotropic, that is to say, no uniformity in all orientations. This normally results into a miscategorisation of pixels. To overcome this problem, Harris and Stephen redesigned the Moravec algorithm by considering differential on corner scores with respect to direction directly. In other words, in the Harris method, changes are captured for the shift in the window in all directions unlike in the Moravec method where only shifts of the window by 45 degree angle are considered. The Harris corner detector method determines the nature of a point by computing the average change of intensity in the image when shifting a small local window in the image by small amount in any direction. Given an image, the following procedure explains how to obtain corner points of the image using Harris corner detection method:

Let us consider a grayscale image with intensities denoted by I. Let w be a window of size (u, v) consisting of a portion of pixels. Suppose the window is shifted by (x, y). The change of intensity due to the shift is given by [102]

$$E(x,y) = \sum_u \sum_v w(u,v)\ [I(u+x, v+y) - I(u,v)]^2, \tag{2.7.1}$$

where $w(u,v)$ is the Gaussian window function $e^{-\frac{u^2+v^2}{2\sigma^2}}$ with parameter σ. The Gaussian function is often used as it ensures isotropic response due to the shift.
Let I_x and I_y be the partial derivatives of I. Using Taylor series expansion, an approximation of $I(u+x, v+y)$ is given as

$$I(u+x, v+y) \approx I(u,v) + I_x(u,v)x + I_y(u,v)y + \text{ higher order terms} \tag{2.7.2}$$

For a small shift (x, y), we eliminate the high order terms as the first order approximation is good enough and thus an approximation of equation 2.7.1 is given by

$$E(x,y) \approx \sum_u \sum_v w(u,v)\ [I_x(u,v)x + I_y(u,v)y]^2, \tag{2.7.3}$$

whose matrix form is

$$E(x,y) \approx \begin{bmatrix} x & y \end{bmatrix} \boldsymbol{M} \begin{bmatrix} x \\ y \end{bmatrix}, \tag{2.7.4}$$

where

$$\boldsymbol{M} = \sum_u \sum_v w(u,v) \begin{bmatrix} I_x(u,v)^2 & I_x(u,v)I_y(u,v) \\ I_x(u,v)I_y(u,v) & I_y(u,v)^2. \end{bmatrix} \tag{2.7.5}$$

The matrix, $\boldsymbol{M}$, is known as the Harris matrix. It describes the shape of the local autocorrelation function E at the origin.

The function $E(x,y) = p$, where p is a constant, is an ellipse with axis lengths and directions determined by eigenvalues and eigenvectors of matrix $\boldsymbol{M}$. The eigenvalues of $\boldsymbol{M}$ are proportional to the principal curvatures of the autocorrelation function E, and form a rotationally invariant description of M: the direction of fastest change is the eigenvector parallel to the minor axis of the ellipse (minimum eigenvalue) and the direction of slowest change is the eigenvector parallel to the major axis of the ellipse (maximum eigenvalue). Moreover, the eigenvalues of $\boldsymbol{M}$ can be used in classification of image points: Let λ_1 and λ_2 be the eigenvalues of $\boldsymbol{M}$. For a "flat" region λ_1 and λ_2 are small hence E is almost constant in all directions. For an "edge", either $\lambda_1 \gg \lambda_2$ or $\lambda_2 \gg \lambda_1$. For a "corner", λ_1 and λ_2 are large, $\lambda_1 \sim \lambda_2$ thus E increases in all directions.

In order to reduce complexity associated with computing eigenvalues, the authors in [57] use the response function, denoted by R, as a measure of corner quality. It is given by

$$R = Det(\boldsymbol{M}) - k(Trace(\boldsymbol{M}))^2, \tag{2.7.6}$$

where $Det(\boldsymbol{M}) = \lambda_1\lambda_2$, $Trace(\boldsymbol{M}) = \lambda_1 + \lambda_2$ and k is an empirical constant such that $k \in [0.04, 0.06]$.

Finally, all corner points with $R > Th$, where Th is the threshold, are selected. This is then followed by computation of non-maximum suppression which involves selecting points of the local maxima of the response function R. All other corner points that do not lie in the local maxima are suppressed to zero.
A clear step-by-step algorithm and Matlab code for the Harris corner Detector can be reviewed [102]. Another response function which does not rely on the value of k was introduced by Noble [4]. It is given by

$$R = \frac{Det(\boldsymbol{M})}{Trace(\boldsymbol{M})} \tag{2.7.7}$$

In summary, the steps for performing Harris corner point detection are as follows:

i) For the input image, compute the derivatives $I_x(u,v)$ and $I_y(u,v)$ of each pixel.

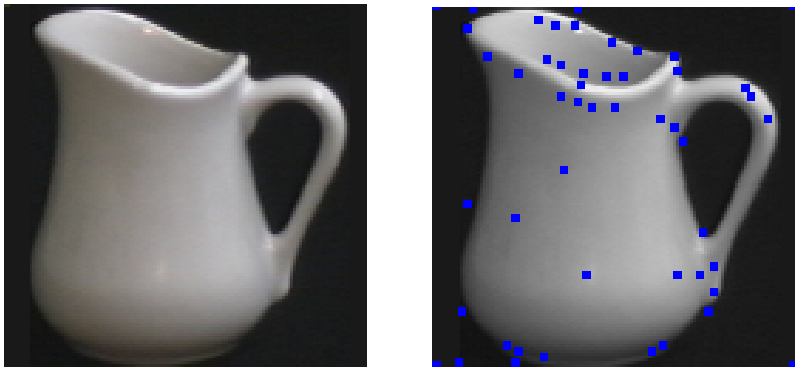
ii) Compute products $I_x(u,v)^2$, $I_y(u,v)^2$ and $I_x(u,v)I_y(u,v)$.

iii) Compute matrix, $\boldsymbol{M}$, in a Gaussian window around each pixel.

iv) Compute corner response function R, that is $R = det(\boldsymbol{M}) - k(Trace(\boldsymbol{M}))^2$.

v) Threshold on R which entails finding points with response function greater than threshold ($R > Th$).

vi) Compute non-maximum suppression which involves selecting points of the local maxima of R and suppress the rest of the points that do not lie in the local maxima to zero.



(a) (b)

Figure 2.11: Illustration of corners of a sample image (Source: [85]) using Harris corner detector. (a) is a sample image and (b) is the image with Harris corner points superimposed.

2.8 Voronoi diagrams and Delaunay triangulation

Voronoi diagrams and Delaunay triangulation are very useful tools in various fields but most commonly in computational geometry.

2.8.1 Voronoi diagrams

Voronoi diagrams (also called Voronoi tessellations, Voronoi decompositions, or Dirichlet tessellations) are important geometrical structures that are found almost everywhere in the world. They have a wide range of applications such as modeling of biological structures such as cells, in pattern-based art work, study of growth patterns of forests and forest canopies in ecology, tracing sources of infections in epidemics, finding clear routes in autonomous robot navigations, among others [91]. Before defining what Voronoi diagrams are, let us first briefly present some of the terminologies we need to know beforehand:

Definition 2.8.2 (Metric space)**.** A metric space is a set E, together with a function d (known as a metric) which operates on each pair $a, b \in E$ such that the following properties hold:

- $d(a,b) \geq 0$, equality holds if and only if $a = b$.
- $d(a,b) = d(b,a)$
- $d(a,b) + d(b,c) \geq d(a,c)$ (known as the triangular inequality)

Let $R = \{x_1, x_2, \ldots, x_n\}$ be a set of n real numbers, an n-dimensional euclidean space is a metric space that consists of set E_n of all n-tuples of R. Let $a, b \in E_n$ such that $a = (p_1, p_2, \ldots, p_n)$ and $b = (q_1, q_2, \ldots, q_n)$ then the function $d(a, b)$ is given by

$$d(a,b) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \ldots + (p_n - q_n)^2}. \tag{2.8.1}$$

Definition 2.8.3 (Voronoi)**.** A Voronoi diagram is a special kind of decomposition of a metric space (or plane) into regions based on distances to a specified discrete set of objects in the space (usually denoted by a set of points normally referred to as seeds, sites or generators), according to the nearest-neighbor rule, such that each point is associated with the region of the plane closest to it [8]. The regions are referred to as Voronoi cells. A Voronoi vertex is the common boundary of 3 adjacent cells. The Voronoi edge, on the other hand, is the common boundary of two adjacent cells.

One simple and practical example of an application of Voronoi diagrams: suppose a particular chain of stores is to be introduced in a given country. The shops have to be established in different locations such that the whole country is covered and with as minimal shops as possible. To address this challenge, suppose we have n shops located at different cities of the country. The different locations of the shops are a subset of points denoted by $S = p_1, p_2, \ldots, p_n$. We assume that the distance between two points is given by the Euclidean distance function of equation 2.8.1 with $n = 2$.
Assuming other factors remain constant and that customers choose to buy from a shop closer to them, then the estimated number of customers to be served by a shop located at spot p_k is given by the region R_k around point p_k.
Let us briefly review some of the definitions used in the study of Voronoi diagrams as by [8]:

Definition 2.8.4 (Convex set)**.** In a Euclidean space, a convex region (or set) is a region where, for every pair of points within the region, every point on the straight line segment that joins the pair of points is also within the region.

Definition 2.8.5 (Convex hull)**.** The convex hull (or convex envelope) of a set P of points in the Euclidean plane or in a Euclidean space is the smallest convex set that contains P.

Definition 2.8.6 (Simplex)**.** A k-simplex is a k-dimensional polytope (a geometric object with flat sides) which is the convex hull of its $k + 1$ vertices. For example 1-simplex is a line segment, 2-simplex is a triangle and 3-simplex is a tetrahedron.

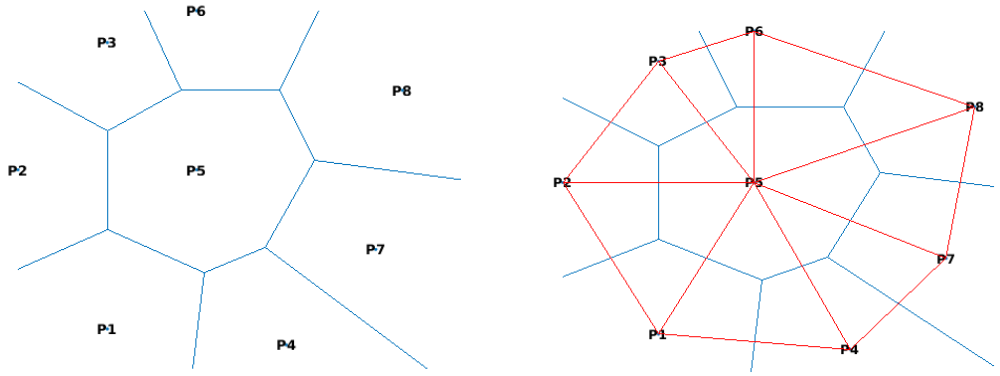
Definition 2.8.7 (Planar graph)**.** A planar graph is a graph that can be drawn in the plane without any edges crossing. A planar graph drawn in this way divides the plane into regions bounded by the edges of the graph. These regions are called faces.

Definition 2.8.8 (Dual graph)**.** The dual graph denoted as G^* of a plane graph G is a plane graph whose vertices correspond to the faces of G and whose edges correspond to the edges of G.

2.8.9 Delaunay triangulation

Let $P = p_1, p_2, \ldots, p_n \subset \mathbb{R}^2$ be a point set. A triangulation $\mathcal{T}$ of P is a maximal planar subdivision with vertex set P. The triangulation $\mathcal{T}$ is referred to as a Delaunay triangulation of P if and only if the circumcircle (a circle that touches each of the three vertices of a triangle) of any triangle in $\mathcal{T}$ does not contain a point of P in its interior [8]. Delaunay Triangulation has been applied in various fields for example climate and global modeling [65], modeling computer vision problems [33], and many others. Most interesting, the minimum spanning tree of a set of points P is a subgraph of the Delaunay triangulation. Following this theorem, the Delaunay triangulation is applied to approximating the Euclidean Traveling Salesperson Problem (ETSP) [46].

There are various methods of computing the Delaunay triangulations which include plane sweeping, iterative flipping from any other triangulation, randomized incremental construction, and conversion from Voronoi diagram [46]. For this work, however, we will consider the Voronoi diagram based method which is founded on the relation that for a euclidean space with point sites, the dual graph of the Voronoi diagram corresponds to the Delaunay triangulation whose vertices are the point sites. The regions are referred to as Voronoi cells. A Voronoi vertex is the common boundary of 3 adjacent cells. The Voronoi edge, on the other hand, is the common boundary of two adjacent cells.



(a) (b)

Figure 2.12: A sample Vorronoi diagram on 8 seeds/points (blue) in (a) and its corresponding dual graph (Delaunay triangulation in red) superimposed in (b).

In chapter 4, we discuss how a combination Harris corner point detection and Delaunay triangulation can be applied to extraction of graphs from objects.

2.9 Principal Component Analysis (PCA)

Principal Component Analysis is a powerful tool for data analysis especially for purposes of dimension reduction in high dimension data. PCA technique is applied in various applications such as image compression, face recognition, pattern identification in high dimensional data, among others [98]. Here, we briefly discuss the core steps involved in performing PCA on a given a data set with n dimensions. Let us take $n = 3$ such that we have dimensions x, y, and z:

- First, we compute the mean for each dimension. For example for dimension x, the mean value, $\bar{X}$, is given by
$$\bar{X} = \frac{\sum_{i=1}^{n} x_i}{n}, \tag{2.9.1}$$
where n is the number of elements in dimension x.
- Next, we obtain normalised data set by subtracting the mean from the respective dimension, for example, for dimensions x and y, subtract each x value, compute $x - \bar{x}$ and for each y value, compute $y - \bar{y}$.
- Let $Cov(x, y)$ denote the covariance between dimensions x and y. We then compute the covariance matrix as matrix is given by
$$\boldsymbol{C}^{m\times m} = (\boldsymbol{C}_{i,j}, \boldsymbol{C}_{i,j} = Cov(Dim_i, Dim_j)), \tag{2.9.2}$$
where $C^{m\times m}$ is a matrix with m rows and m columns, Dim_i is the i-th dimension. For example, for a data set with 3 dimensions x,y, and z, we write the covariance matrix as
$$\boldsymbol{C} = \begin{pmatrix} Cov(x,x) & Cov(x,y) & Cov(x,z) \\ Cov(y,x) & Cov(y,y) & Cov(y,z) \\ Cov(z,x) & Cov(z,y) & Cov(z,z) \end{pmatrix} \tag{2.9.3}$$
- Compute the eigenvalues and eigenvectors of the covariance matrix.
- For dimensional reduction and data compression, choose principal components which are the eigenvectors corresponding to largest eigenvalues. The principal components are the directions where there is most variance (that is, where data is most spread). The number of principal components corresponds to the new dimensionality.
- Feature vector formation. Having chosen which eigenvectors to keep, we construct the feature vector which is a matrix with the chosen eigenvectors as columns.
- New data set. We obtain the new data set (with reduced dimensions) by
$$NewData = FeatureVector^T \times NormalisedData^T \tag{2.9.4}$$

Having obtained the feature vectors as explained in the previous subsection, we create a database whose columns are the feature vectors for each graph. We can then apply principal components analysis to the database so as to embed the respective graphs onto low dimension space as we will elaborate in chapter 4.

2.10 Conclusion

In this chapter, we have reviewed relevant literature which includes a brief introduction to and terminology used in the study of graphs and networks. We further discussed the network approach to the study of systems; first, we looked at how networks can be used to represent artificial and real-world systems with nodes representing components that constitute the system while links represent interconnections between components. Second, we discuss the use of different types of matrices to represent networks or graphs. Matrices are good representation of networks and aid in compuations related to networks. More so, we discuss the structure of graphs/networks and some measures used in characterising the structure of a graph or network. We note that though useful insights about network structure can be obtained using these measures, in some cases, however, networks with different structures can have the same value of a particular measure.

In addition, we briefly discuss degree distributions and random models of networks, particularly focusing on the common models namely ER and BA which are used to design networks depicting both artificial and real-world networks. We further lay a foundation for performing object clustering using graph-based approach (discussed in later chapters) by reviewing the following concepts; graph similarity, corner point detection for images, Voronoi diagrams, Delaunay triangulations, and Principal Component Analysis.

Chapter 3

Diffusion on networks

In the previous chapter, we reviewed some of the relevant literature. In this chapter, we will explore the diffusion process on networks. First, we look at the model where diffusion occurs along the edges of a network. We also study the equilibrium behaviour under this model and also investigate the role played by the structure of a network, choice of source nodes, and network homogeneity on the diffusion process on a network. Moreover, we explore diffusion on directed networks and discuss equilibrium behaviour on different types of directed networks. Furthermore, we consider an extension of the above model introduced in [38, 39]. Here, diffusion on networks involves both direct interactions (among nearest neighbor nodes) as well as long-range interactions between pairs of non-nearest nodes in the network. Our interest in this model is the significant contribution of long-range interactions observed in many processes in real-world for example epidemic spread in population, synchronisation processes in physics and consensus problems, to mention but a few.

3.1 Introduction

Diffusion on networks is a particular example of a broad concept of dynamical systems on networks. A dynamical system is defined as a system whose state, as represented by a given set of quantitative variables, changes over time according to some given rules or equations [86]. Diffusion on networks aid in developing simplified models of real-world processes that occur in real-world systems for example propagation of infection within a population, transmission of signals in brain networks, dissemination of information in a social network, virus spread in computer network, spread of money within an economy, and many others [69].

In network theory, diffusion is a process by which behaviours such as information, heat, viruses spread over networks [69]. In the study of diffusion process in networks, a familiar model discussed in literature is based on the idea that the behaviour under study spreads through a network along paths that connect pairs of nodes. This model has been widely used in modeling the spread of information, ideas, opinions in social networks, social media marketing of commodities and infection spread among population [13, 41, 69, 76].

Estrada [41] pointed out that in modeling the spread of epidemics, the most important task is the determination of a network of social contacts over which the spread of infections occurs. Though it is an easy task in cases such as spread of sexually transmitted diseases, it is however a challenging task in cases that involve spread of airborne or close contact infections [41]. This is so because the airborne or close contact infections can be transmitted in two ways: First, through interactions between individuals connected to each other in the social network known as close contacts. Second, through encounters between two individuals that are not close to each other in the network. These contacts are known as casual contacts and they play a vital role in the spread of infection as evidenced in the case of severe acute respiratory syndrome (SARS) discussed in [100], spread of obesity [28], among others. It is thus of utmost importance to account for the casual contacts when modeling the spread of infections. To that cause, various experiments have been carried out to account for both close and casual contacts. An interesting attempt by Estrada et al. [41] to account for the two types of contacts among individuals was by considering the transmission of an infection through paths (direct interactions) and by long-range (LR) interactions in a complex network. This method is based on the idea that with the structure of network of close contact to hand then the casual contacts can be deduced for any pair of individuals as long range interactions based on the distance of separation between them. It is worth noting that in this model, the long-range interactions are considered to be nonrandom as opposed to other models in which casual contacts were assumed to occur randomly.

3.2 The standard diffusion model

The most common model used in the study of diffusion on networks considers the spread of behaviour or substance along the edges from vertices with a high value to vertices with lower values in the network. In this section, we describe the diffusion process on a network based on this model. Let $G = (V, E)$ be a simple connected undirected graph with vertex set V and edge set E. Let the state variable, $\phi_i(t)$, be the amount of heat at vertex, i, at time, t, and $\varepsilon \in [0, 1]$ be the heat diffusion coefficient that controls the rate of diffusion. When ε tends to 0, heat transfer among nodes becomes difficult and as a result, heat does not spread to each of the nodes with in the network. However, as ε tends to 1, heat spreads rapidly among nodes and thus, with out loss, heat is distributed to all nodes in the network. Suppose that the vector $\boldsymbol{\phi}_0$ describes the initial heat distribution over nodes of a network and heat flows along the edges connecting nodes, that is to say, through direct interactions, then the process of heat spread through the network can therefore be modelled by

$$\frac{d\phi_i}{dt} = \varepsilon \sum_j \boldsymbol{A}_{ij}(\phi_j(t) - \phi_i(t)) = \varepsilon \sum_j (\boldsymbol{A}_{ij} - \delta_{ij} k_i)\phi_j(t) \tag{3.2.1}$$

$$= -\varepsilon \sum_j \boldsymbol{L}_{ij}\phi_j(t), \tag{3.2.2}$$

where $\boldsymbol{A}$ is the adjacency matrix, k_i is the degree of node i, and δ_{ij} is the Kronecker delta whose value is 1 if $i = j$ and 0 otherwise. In matrix-vector notation, we have

$$\frac{d\boldsymbol{\phi}}{dt} = -\varepsilon \boldsymbol{L}\boldsymbol{\phi}(t), \quad \boldsymbol{\phi}(0) = \boldsymbol{\phi}_0, \tag{3.2.3}$$

where $\boldsymbol{L}$ is the Laplacian matrix of a graph. Equation 3.2.3 has the form of the heat equation with the Laplacian operator ∇^2 replaced by $-\boldsymbol{L}$ which is the reason why $\boldsymbol{L}$ is called the graph Laplacian. Alternatively, the normalised version $\boldsymbol{\mathcal{L}}$ of the Laplacian can is used.
The solution to equation 3.2.3 is

$$\boldsymbol{\phi}(t) = \boldsymbol{\phi}_0 \, e^{-\varepsilon t \boldsymbol{L}}. \tag{3.2.4}$$

Alternatively, by using the spectral decomposition of the Laplacian, equation 3.2.4 can be written as

$$\boldsymbol{\phi}(t) = \sum_i \boldsymbol{\phi}_0 \cdot \boldsymbol{v}_i \, e^{-\varepsilon \lambda_i t} \boldsymbol{v}_i, \tag{3.2.5}$$

where λ_i, $\boldsymbol{v}_i$ are respectively the eigenvalues and corresponding eigenvectors of the Laplacian matrix and $\boldsymbol{\phi}_0 \cdot \boldsymbol{v}_i$ is simply the projection of $\boldsymbol{\phi}_0$ onto the set of eigenvectors [7].

3.3 Equilibrium behaviour

For any dynamical system, equilibrium is a very vital concept. The equilibrium is the simplest possible solution to a dynamical system. It is a solution where the state variables do not change with time. Most importantly, the study of equilibrium lays a foundation for analysing the behaviour of a system when time goes to infinity. In this section therefore, we study the equilibrium of diffusion process on a simple undirected network. As t goes to infinity, we have

$$\lim_{t\to\infty} e^{-\varepsilon\lambda_i t} = \begin{cases} 0 & \text{if } \lambda_i > 0 \\ 1 & \text{if } \lambda_i = 0. \end{cases} \tag{3.3.1}$$

Asymptotically, the equilibrium state is completely determined by the kernel of $\boldsymbol{L}$. Since $\sum_j \boldsymbol{L}_{ij} = 0$, it is easy to see that $\boldsymbol{v}^1 = \frac{1}{\sqrt{n}}[1, 1, \ldots, 1]$, the eigenvector associated with $\lambda_i = 0$, is in the kernel of $\boldsymbol{L}$. We then have

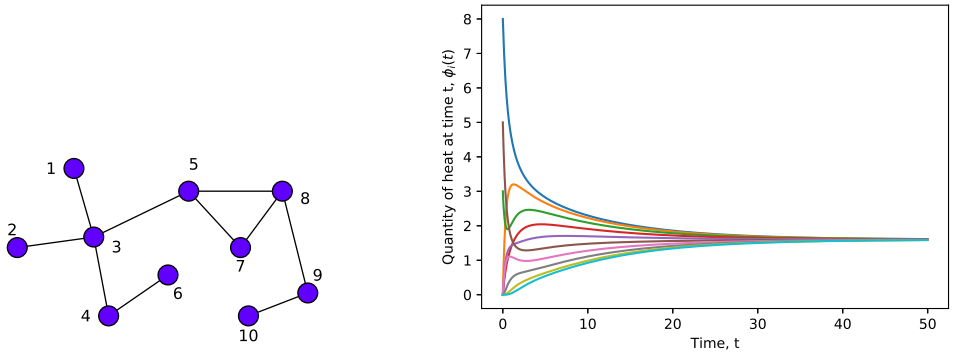
$$\lim_{t\to\infty} \boldsymbol{\phi}(t) = (\boldsymbol{\phi}(0) \cdot \boldsymbol{v}^1)\boldsymbol{v}^1. \tag{3.3.2}$$

The quantity of heat $\boldsymbol{\phi}_j(t)$ at any node j at time t is given by

$$\lim_{t\to\infty} \boldsymbol{\phi}_j(t) = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\phi}_i(0). \tag{3.3.3}$$

At steady state, the quantity $\boldsymbol{\phi}_i(t)$ at each of the nodes is the same. It is the average of the initial values at all of nodes. This is because, as expected, neighboring nodes in the network will exchange heat amongst each other until all nodes attain equal amounts of heat (that is, no difference in heat quantities for any given pair of nodes).
For better understanding of the concept of equilibrium behaviour on networks, let us consider a simple example:

Example 3.3.1. Let us consider diffusion of heat over the network in Figure 3.1a. Suppose the quantity of heat at each node at time $t = 0$ is given by the vector $\boldsymbol{\phi}(0) = [8, 0, 3, 0, 0, 5, 0, 0, 0, 0]$ of random values between 0 and 10. Let $\varepsilon = 0.05$. Figure 3.1b illustrates how heat spreads over the network in Figure 3.1a. In this thesis, most simulations of diffusion on networks are implemented using NETWORKX in PYTHON [55].



(a) (b)

Figure 3.1: Results from simulation of diffusion process over the network in (b).

From Figure 3.1, we observe that at each time step t, nodes that initially have high amounts of heat (that is to say, 1, 3, and 6) exchange heat with adjacent nodes that initially had none or little amounts of heat and eventually all nodes in the network have relatively equal amounts of heat. As observed in Figure 3.1b, with time the quantity of heat $\boldsymbol{\phi}_j(t)$ at each node tends to equilibrium value of 1.6 which is attained at about $t = 50$.

Following from equation 3.3.3, it is clear that at equilibrium, the state variables at each of the node has the same value which is the average value of the initial values at all nodes in the network. It should, however, be noted that equilibrium value is not always the average value of the initial distribution for instance in some directed networks, equilibrium can be attained at different values (not the average of initial values) for each of the nodes as discussed in [86].

3.4 Impact of network structure, choice of initial nodes and heterogeneity on diffusion on a network

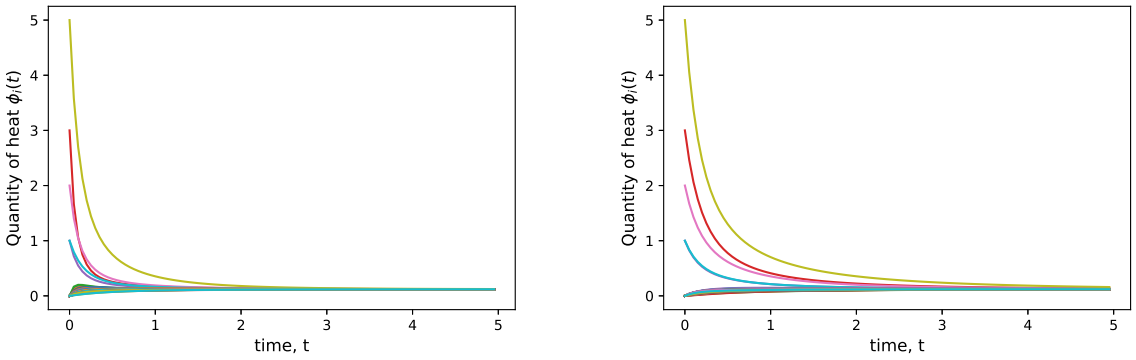
In this section, we explore some of the factors that affect the rate of diffusion on networks.

3.4.1 Impact of network structure on the dynamics of diffusion process on a network

As discussed earlier on in chapter 2, the structure of a network is the way in which nodes are connected in the network. For example, in a regular network each node is connected to equal number of nodes, for a star network one node is positioned in a way that all other

nodes are connected to it. Since diffusion on networks occurs through interactions between neighbouring nodes, it is reasonable to investigate the impact of network structure on the diffusion process on a network.

We consider two networks with different structures: one is an Erdős-Rényi (ER) network that follows a Poisson degree distribution and the other Barabási-Albert (BA) network that follows scale free power-law degree distribution, that is to say, the probability of finding a node with degree k decreases as the negative power of k. We perform simulations to illustrate diffusion on two different networks and the results are explained: Consider ER and BA networks both of size 100 each; we randomly select 5 nodes to which we assign random quantities (in the interval $[1-10]$) of heat and then allow diffusion to occur. After every time step t, we compute the quantities at each node as depicted in Figure 3.2.



(a) (b)

Figure 3.2: Results of the simulations for diffusion on networks using equation 3.2.3. The left hand panel corresponds to BA network and right hand side panel corresponds to ER. Both networks are of size $n = 100$ and average degree $\bar{k} \approx 4$.

In Figure 3.2, we observe that diffusion occurs much faster in BA network than in ER network based on the time at which equilibrium is attained. In BA (see Figure 3.2a), equilibrium is reached after about 2.5 time steps while in ER network (see Figure 3.2b) equilibrium is reached much later after about 5.0 time steps which is double the time in BA network. This behaviour can be attributed to the difference in structure of the two networks. The BA network is composed of highly connected nodes, known as hubs, that interact with many nodes to which they are connected in each time step thus hastening the diffusion process. On the other hand, for the ER network, most nodes are connected to few nodes which explains the slower diffusion process compared to the previous case. From the results, we can thus appreciate that the structure of a network plays a significant role in the diffusion process and thus a determinant of how fast equilibrium can be attained.

3.4.2 Impact of choice of initial diffusion nodes on the diffusion process on networks

Apart from network structure, another factor that affect the diffusion process on a given network is the choice of nodes from which the diffusion process commences. Some nodes of a network are positioned in such a way that they can quickly interact with many other nodes in a network and when selected as source nodes to which initial qualities of heat greater than zero are assigned, diffusion can occur faster. To throw more light on this concept, we consider an ER network and BA network of 1000 nodes. First, we randomly select 10 of the nodes in both networks to be source nodes. Second, we select 10 nodes with the highest degree centrality to be considered as source nodes. In both cases, we assign initial quantities of heat obtained as random values in the interval $[1, 100]$ to the 10 source nodes while the rest are assigned values of zero. We then obtain quantities at each node at time step t. The results of the simulations are illustrated in Figure 3.3.

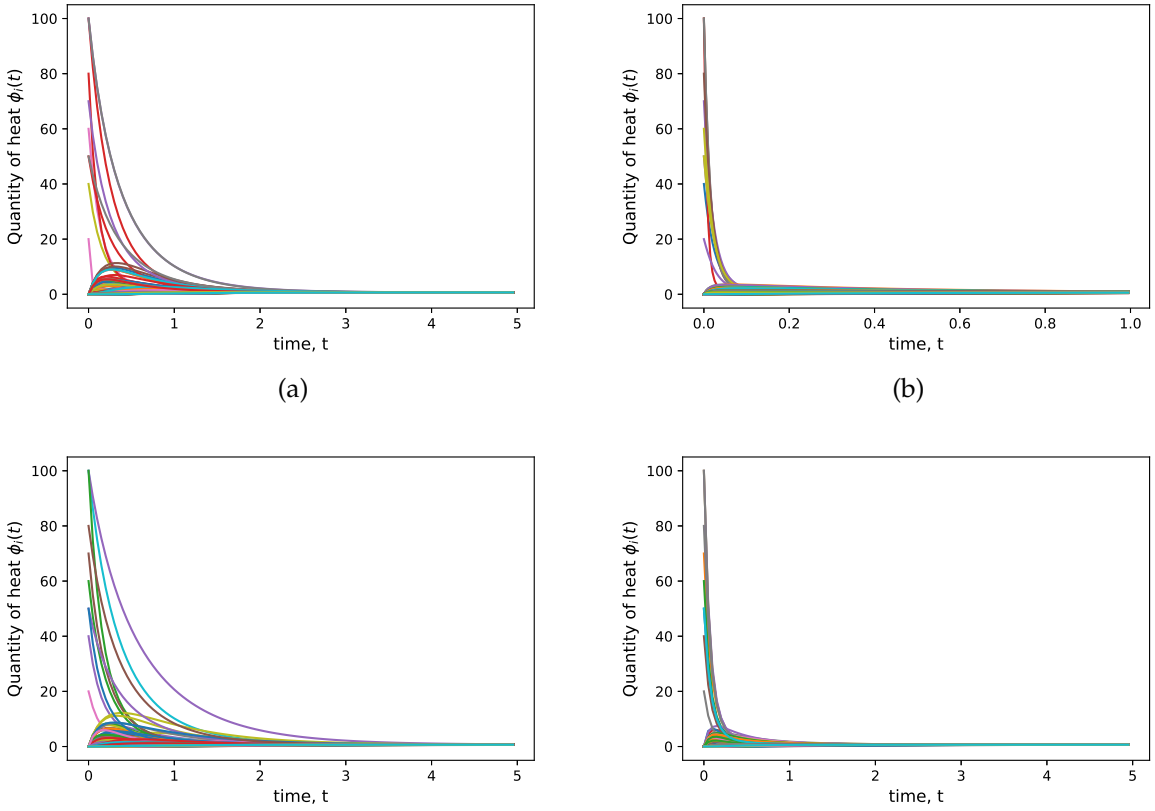


Figure 3.3: Results of the simulations for diffusion on BA (top row) and ER (bottom row) networks both of size $n = 1000$ and $\bar{k} \approx 6.0$. The left column corresponds to the case where source nodes are selected randomly while the right hand column corresponds to the case for which highest degree nodes are chosen as source nodes.

Figure 3.3 illustrates the results of simulations of diffusion of heat on BA and ER networks

under conditions of choice of nodes from which diffusion commences, that is to say, nodes to which quantities of heat greater than zero are assigned at $t = 0$. First, we consider a scenario in which source nodes are selected randomly. Second, we consider selection of highest degree nodes as source nodes. In the first case (Figures 3.3a and 3.3c), we observe that diffusion occurs slightly faster in BA network (see Figure 3.3c) than in ER network (see Figure 3.3a). We explain this observation based on the fact that since source nodes are selected randomly in both networks, there is a high possibility of selecting mostly low degree nodes than high degree nodes (hubs) in the BA network since hubs are very few and thus the probability that such nodes are selected as source nodes is very low. When low degree nodes are selected as source nodes, the rate of diffusion is relatively slow as the flow of substance between nodes occurs in a few nodes at a time. On the other hand, when highest degree nodes are selected as source nodes as shown in Figures 3.3b and 3.3d for BA and ER networks respectively, we observe a drastic increase in the rate of diffusion in both the ER and BA networks compared to the previous case. However, we can see that the diffusion process is much faster in BA network than in ER network as supported by the times at which equilibrium is attained, that is, equilibrium is attained at about $t = 0.8$ while in ER network, equilibrium is attained at $t = 2.5$. This huge difference in equilibrium times can be explained based on the fact that in BA networks, there exists high degree nodes called hubs and when such nodes initiate the diffusion process, they quickly spread the heat to many other nodes to which they are connected in a short period of time which results into faster diffusion. For ER networks, almost all nodes have the same and relatively low degree and thus, when some of these nodes are selected to initiate diffusion, for each time step, they spread heat to their few neighbouring nodes (compared to hubs) which makes the diffusion process slower in this case than in the former case. The concept of the impact of the choice of initial selection of nodes very important as it can be used in applications to control the flow of information or any other substance within a system.

3.4.3 Influence of heterogeneity on diffusion over network

In literature, the heterogeneity of a network has been defined in two aspects as based on analysis made by authors in [63]. In [3, 14], heterogeneity is defined as the diversity in node degrees. According to authors in [2, 36, 87], heterogeneity of a network is the irregularity characterised by the existence of nodes with degree significantly larger than the average degree of the network. The star network is considered as the most heterogeneous for which the average degree tends to 2 as the number of nodes increases. On the other hand, the regular network is considered the least heterogeneous as all nodes have the same degree, that is to say, regular networks are homogeneous.

From the degree distribution, we can obtain some useful insights about the heterogeneity of a network. However, this technique may not perform well in some cases for instance where more than one degree distribution fits for a given network, a case of comparison between two networks with totally different degree distributions, among others. To work around such

cases, various measures that uniquely quantify network heterogeneity have been put forward, for example in [3, 14, 36].

For simulations, we consider heterogeneity in scale free networks by varying power exponent, γ using the method proposed in [111]. We consider two networks with $\gamma = 2.0$ and $\gamma = 2.3$. both networks consists of 1000 nodes each. We commence by assigning random initial quantities of heat to each of the 10 nodes with highest degree while the rest of the nodes are assigned values of zero. This forms the initial heat distribution in the networks. Figure 3.4 illustrates results of simulations of diffusion over the networks with time.

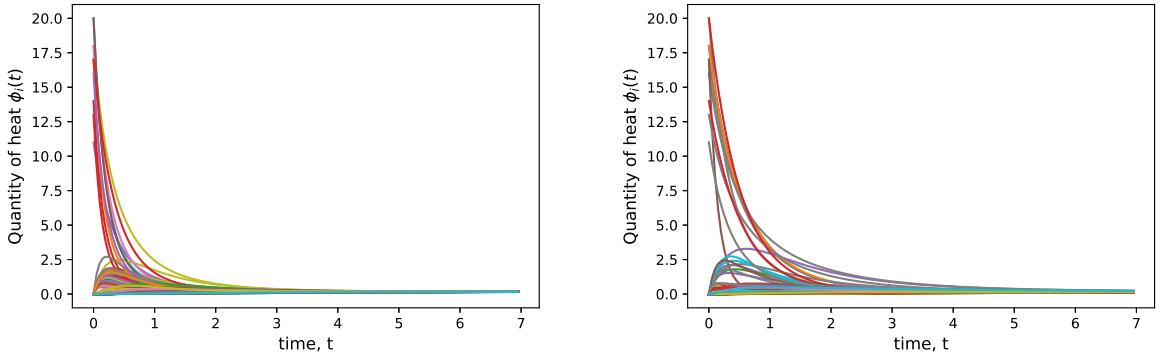


Figure 3.4: Results of simulation on two scale free networks with power exponents 2.0 (left) and 2.3 (right). Both networks have average degree $\bar{k} = 5.4$ and $\bar{k} = 5.2$ respectively.

We consider two scale free networks with γ values of 2 and 2.3. On simulating diffusion of heat on the two networks where 100 high degree nodes are taken as initiators of the spread, we observe in Figure 3.4 that diffusion occurs much faster in the more heterogeneous network ($\gamma = 2.0$) than in its less heterogeneous counterpart ($\gamma = 2.3$). This is because, in the former there are more nodes with very high degree (hubs) and numerous nodes with very low degree and this large difference in degree is responsible for the faster spread of heat since hubs form part of source nodes and thus reaches many other nodes with in a short time. On the other hand, the latter network is more homogeneous and thus few nodes with degree far above the average. Though highest nodes are selected as source nodes, they are not as densely connected to the rest of the nodes as in the previous case and so relatively few nodes are reached in each time step resulting into a slower diffusion process.

3.5 The generalised diffusion model

3.5.1 Motivation

Many real-world systems consist of an interconnection of different components. The collective dynamic behaviour or functionality of these systems is, to a large extent, determined by the interactions between constituent components and thus studying an individual component in isolation cannot guarantee understanding of a system's functionality but rather a holistic view of the system is required. Recently, researchers from various fields such as physics, sociology, and mathematics have been deeply involved in complex network related research and various applications of complex networks in these areas have been documented.

In network science, complex networks have been used to represent structures of complex systems. This representation has aided in understanding and predicting the dynamic behaviour of real-world systems using the network approach. The network approach involves representing the system by a network, followed by the studying the dynamical interactions between nodes (represent components). The majority of such models are based on the idea that interaction between nodes only occurs along the edges, that is to say, nodes interact with only their nearest neighbours (nodes with which they share an edge) as discussed in section 3.2. For example, in modelling the spread of sexually transmitted diseases, interactions between nodes (individuals) are considered to occur along edges of the social network of close contacts. However, in many natural and artificial dynamical processes such as the spread of infections among people, consensus process in multi-agent systems, synchronisation dynamics, among others, it has been observed that interaction between constituent components occurs via both direct interactions between nearest components as well as interactions between non-nearest components which are referred to as long-range interactions [41].
For example, in the account of the spread of the severe acute respiratory syndrome in China [100], the authors pointed out the fact that apart from the infections passed on from one person to another with in china, the disease happened to spread to other countries such as Viet Nam, Canada, and Singapore as a result to interactions between a medical doctor from China and people from the respective countries during their stay in a hotel in Hong Kong. The people from the mentioned countries other than china are not close contacts to the doctor. In addition, Estrada [38] indicated that in many real-world networks that consist of highly connected clusters which are poorly linked amongst themselves (known as communities), it is expected that consensus with in a particular community is reached faster and as a result consensus can be attained easily for different communities. On the other hand, consensus between nodes belonging to different communities is quite slower which implies that overall consensus with the network is reach slowly. However, it is observed that in such networks, consensus is reached much faster than anticipated. According to Estrada, this behaviour could possibly be justified by the interaction between nodes from different communities. Such interactions between non-neighbouring nodes are referred to as long-range interactions and are considered to be nonrandom [38].

A number of models that capture these interactions have been presented in literature for example, Random Walks with Levy Flights (RWLF) [71], Fractional Diffusion Equation (FDE), k-path Laplacian based model [40], and many others. In this work, however, we adopt an elegant model based on k-path Laplacian matrices introduced by Estrada [38]. To start with, we briefly review the concept of k-path Laplacian matrices.

3.5.1.1 k-path Laplacian matrices, L_k

The k-path Laplacian matrices are a natural generalisation of the combinatorial Laplacian of a graph [38]. The motivation behind this generalisation is the idea of determining whether every node in a graph can be visited by means of a process that involves hopping from one node to another separated at a distance k. We can better understand the concept of the k-path Laplacian by considering the analogy of polarisation process on a network put forward by Estrada [38]: Suppose a particle with a positive charge resides at a given node of simple graph $G = (V, E)$ and while at that node, it polarises all nodes at a distance k from it. Consequently, the particle's movement from the current node is guided by the rule that the particle can only to any nearest non-positively charged particle. While at the new node, the particle polarises neighbouring nodes in the same manner as before. As a result, the particle either hops to the nearest non-positive nodes or returns to the origin node. This process goes on until all nodes are visited or when further hopes result into a loop. Figure 3.5 illustrates a possible hope on a simple graph.

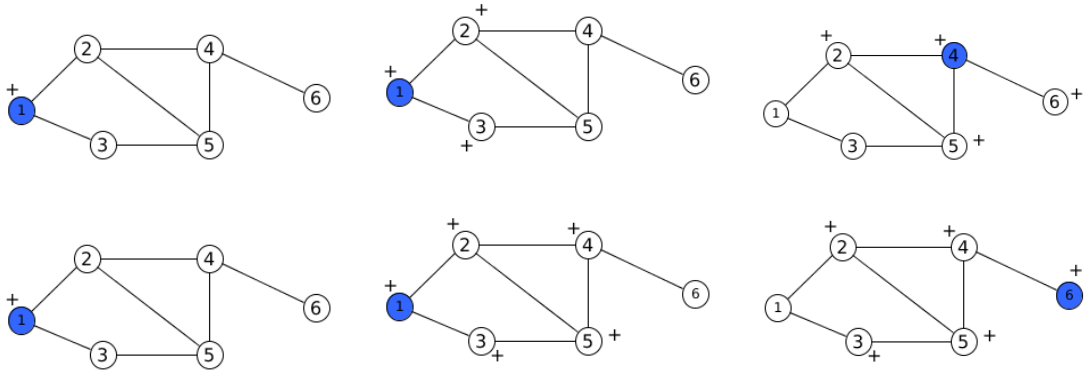


Figure 3.5: Illustration of how the polarisation analogy used as a motivation for the k-path Laplacian concept for networks. Suppose a particle starts off at node labelled 1, top row and bottom row correspond to possible pattern traversed by the particle for $k = 1$ and $k = 2$ respectively

.

For the standard case in which traversing the graph involves subsequent hops of length 1 at a time, terminology such as walk, path, and many more are defined in chapter 2. In the same way, for the generalised case in which hops of various length not exceeding the diameter of a

graph are taken into account, (that is to say, $1 < k \leq d_{max}$, where d_{max} denotes the maximum diameter of a network), the following terminology have been defined in [38]:

Definition 3.5.2 (k-hopping walk)**.** A k-hopping walk of length l is any sequence of (not necessarily different) nodes $v_1, v_2, \ldots, v_l, v_{l+1}$ such that $d_{i,i+1} = k$ for each $i = 1, 2, \ldots, l$. In other words, this walk is referred to as a k-hopping walk from v_1 to v_{l+1} [38].

Definition 3.5.3 (k-path degree)**.** The k-path degree $\delta_k(v_i)(k \leq d_{max})$ of a node v_i is the number of irreducible shortest-paths of length k having v_i as an endpoint [38].

For a simple graph in Figure 3.6 with diameter equal to 2, the k-degree ($k \leq 2$) for each vertex is summarised in Table 3.1.

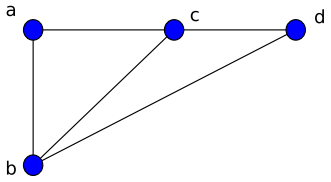


Figure 3.6: A simple graph.

Vertex	δ_1	δ_2
a	2	1
b	3	0
c	3	0
d	2	1

Table 3.1: The k-path degrees for vertices of graph in Figure 3.6.

Definition 3.5.4 (k-path Laplacian matrix)**.** The k-path Laplacian matrix $\boldsymbol{L}_k(k \leq d_{max})$ of a connected undirected graph $G = (V, E)$ is defined as the square symmetric $n \times n$ matrix whose entries are given by

$$\boldsymbol{L}_k(ij) = \begin{cases} \delta_k(i) & \text{if } i = j, \\ -1 & \text{if } d_{i,j} = k, \\ 0 & \text{otherwise,} \end{cases} \tag{3.5.1}$$

where $d_{i,j}$ is the shortest path distance between nodes i and j, $\delta_k(i)$ is the k-path degree of node i [38].

This concept has been extended to connected and locally finite infinite graphs [40].
For clarity, let us compute the k-path Laplacian matrices for the simple graph in Figure 3.6. Note that since $d_{max} = 2$ for the graph, then we consider $k = 1$ and $k = 2$ and thus we have:

$$\boldsymbol{L}_1 = \begin{pmatrix} 2 & -1 & 0 & -1 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ -1 & -1 & -1 & 3 \end{pmatrix}, \quad \boldsymbol{L}_2 = \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

In [38], Estrada proved that the k-path Laplacian matrices are positive semi-definite and satisfy the condition:

$$\boldsymbol{y}^T \boldsymbol{L}_k \boldsymbol{y} \geq 0, \tag{3.5.2}$$

where $\boldsymbol{y}$ is an eigenvector of $\boldsymbol{L}_k$ corresponding to eigenvalue $\lambda_1(\boldsymbol{L}_k) = 0$.

Definition 3.5.5 (k-hopping connected component)**.** A k-hopping connected component of a graph $G = (V, E)$ is a subgraph $G' = (V', E')$, $V' \subset V$, $E' \subset E$, such that there is at least one k-hopping walk that visit every node $v_i \in V'$.

As mentioned earlier, the motivation of the generalisation of graph Laplacian is to ascertain whether a given graph can be k-hopped and if not, how many k-hopping connected components exist. The author in [38] showed that the number of k-hopping connected components, denoted by $\eta_k(G)$, in a connected undirected graph $G = (V, E)$ is equal to the multiplicity of zero as an eigenvalue of the k-path Laplacian matrix. Mathematically, $\eta_k(G) = m[\lambda_1(\boldsymbol{L}_k) = 0]$.

Example 3.5.6. Let us consider the graph, G in Figure 3.6, since $d_{max} = 2$ we compute the 1-hopping and 2-hopping connected components of G.

		no. of components	components
$\lambda_i(\boldsymbol{L}_1)$	**0** 2 4 4	1	1- 2- 3- 4
$\lambda_i(\boldsymbol{L}_2)$	**0** **0** **0** 2	3	1-3, 2, 4

Table 3.2: The k-connected components of the graph in Figure 3.6.

We see from Table 3.2, that when we consider hops of length $k = 1$, there is only one connected component which is equal to the multiplicity of 0 as an eigenvalue in the spectrum given as $0, 2, 2, 4$. On the other hand, however, when we consider hops of length 2 ($k = 2$), we obtain 3 components which are reflective of the multiplicity of 0 eigenvalue in the respective spectrum $0, 0, 0, 2$. For details about the k-path Laplacian matrices, we refer the reader to [38].

3.5.7 The generalised Laplacian matrix

The generalised Laplacian matrix is obtained as a linear combination of the k-path Laplacian matrices and it is given by

$$\boldsymbol{L}_G = \sum_{k=1}^{\Delta} c_k \boldsymbol{L}_k, \tag{3.5.3}$$

where $1 \leq \Delta \leq d_{max}$ (diameter of a network) and c_k are the coefficients [38]. The coefficients c_k play a crucial role in the generalisation of diffusion process on network and so determining the values of these coefficients is an important task. The values of c_k are expected to give more weight to shorter than to the longer range interactions. In [38], Estrada proposed two approaches categorised as social and physical ways of influence.

3.5.8 Choice of coefficients, c_k

In [38], Estrada pointed out that in many artificial and natural dynamical systems that communication among the agents of the system follows a spatial decay. For example, in sensor systems, the value of the noise-signal ratio reported by the sensors decreases with increase in the distance between target and the sensor due to attenuation of signal energy. Another example is the aftermath of earth quake where areas close to the epicentre (origin of an earth quake) are heavily affected compared to those far away. In ecological systems, organisms tend to rise and fall in abundance and in synchronisation with other populations of species, (known as spatial synchrony) it is been observed that this behaviour follows a spatial decay with increase in distance. The spatial decay takes on the form $d^{-\alpha}$, where d is the distance from the target, epicentre of an earthquake and separation between species for the three cases respectively. Following from the physical scenarios, a similar idea was adopted by authors in [38] in developing a method to account for long-range interactions between a given pair of nodes separated at a distance k (k is the length of the shortest path between them) whereby weights are assigned based on the fact that the longer the distance of separation, the weaker the influence of long-range interactions. In this thesis, we discuss two approaches to accounting for long-range interactions namely the Laplace transform and Mellin transform-based approaches put forward in [40]:

a) **Laplace transform**

Here, the rate at which long-range influence weakens with increase in distance d that is to say, the decay rate is exponential. Thus, using the Laplace transforms of the k-path Laplacian matrices, we have

$$\boldsymbol{L}_G = \boldsymbol{L} + \sum_{k=2}^{\infty} e^{-\lambda k} \boldsymbol{L}_k. \tag{3.5.4}$$

Thus, in this case the coefficients of equation 3.5.3 are $c_1 = 1$ and $c_{k\geq 2} = e^{-\lambda k}$.

b) **Mellin transform**

In this case, the decay rate follows a power law, that is, k^{-s} where $s > 0$. The generalised Laplace matrix can be computed from the Mellin transform of the k-path Laplace matrices and is given by

$$\boldsymbol{L}_G = \sum_{k=1}^{\infty} k^{-s} \boldsymbol{L}_k, \tag{3.5.5}$$

For this case, the coefficients of equation 3.5.3 are $c_k = k^{-s}$. In [40], it is shown that normal diffusion occurs only when $s > 3$. On the other hand, superdiffusion occurs when $1 < s < 3$.

In summary, given a finite graph $G = (V, E)$ for which both direct interactions and long-range interactions are taken into account. We account for the latter using two models, namely

the Laplace and Mellin transforms of the path Laplacian matrix from which the generalised Laplacian matrix, $\boldsymbol{L}_G$, is derived. It is thus given by

$$\boldsymbol{L_G} = \tilde{\boldsymbol{L}}_\tau = \boldsymbol{L} + \sum_{k=2}^{\Delta} e^{-\lambda k}\boldsymbol{L}_k, \text{ for } \tau = \text{ Laplace}, \lambda > 0, \tag{3.5.6}$$

or

$$\boldsymbol{L_G} = \tilde{\boldsymbol{L}}_\tau = \sum_{k=1}^{\Delta} k^{-s}\boldsymbol{L}_k, \text{ for } \tau = \text{ Mellin}, s > 0. \tag{3.5.7}$$

where λ and s are positive constant parameters for the Laplace and Mellin transforms respectively and $1 \leq \Delta \leq d_{max}$ where d_{max} is the diameter of G.

3.5.9 Properties of the generalised Laplacian matrix

- The generalised matrix $\boldsymbol{L}_G$ is real and symmetric which follows from the fact that $\boldsymbol{L}_G$ is a linear combination of real and symmetric k-path matrices.

  Proposition 3.5.10. The generalised Laplacian matrix, $\boldsymbol{L}_G$, is positive semi-definite, that is, $\boldsymbol{y}^T\boldsymbol{L}_G\boldsymbol{y} \geq 0, \quad \forall \boldsymbol{y} \in \mathbb{R}^n$.

  Proof. Let $\{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n\}$ be an orthogonal basis of $\mathbb{R}^n$.

  $$\boldsymbol{y}^T\boldsymbol{L}_G\boldsymbol{y} = \boldsymbol{y}^T(c_1\boldsymbol{L}_1 + c_2\boldsymbol{L}_2 + \ldots + c_\Delta\boldsymbol{L}_\Delta)\boldsymbol{y} = c_1\boldsymbol{y}^T\boldsymbol{L}_1\boldsymbol{y} + c_2\boldsymbol{y}^T\boldsymbol{L}_2\boldsymbol{y} + \ldots + c_\Delta\boldsymbol{y}^T\boldsymbol{L}_\Delta\boldsymbol{y} \tag{3.5.8}$$

  We know that $\boldsymbol{y}^T\boldsymbol{L}_k\boldsymbol{y} \geq 0$ [38], $c_k > 0$ and $1 \leq k \leq \Delta$. Substituting for $\boldsymbol{y}^T\boldsymbol{L}_k\boldsymbol{y}$ in equation 3.5.8 gives

  $$\boldsymbol{y}^T\boldsymbol{L}_G\boldsymbol{y} \geq 0 \tag{3.5.9}$$

  □

- Like for the standard Laplacian matrix, zero is always an eigenvalue of $\boldsymbol{L}_G$ with eigenvector, $\mathbf{1}$, which is an all ones vector [37].

- Behaviour of eigenvalues with change in s (or λ). As discussed in [39], for a graph $G = (V, E)$ with generalised Laplacian matrix, $\boldsymbol{L}_G$, obtained from either the Laplace or Mellin transform of k-path Laplacian matrices, its eigenvalues behave as follows:

  $$\left(\frac{\mu_i}{N}\right) \longrightarrow \begin{cases} \left(\frac{\lambda_i}{N}\right), & \text{if } \lambda, s \longrightarrow \infty \\ 1, & \text{if } \lambda, s \longrightarrow 0, \end{cases} \tag{3.5.10}$$

  where μ_i and λ_i are the eigenvalues (except the zero eigenvalue) of $\boldsymbol{L}_G$ and $\boldsymbol{L}$ respectively and N is the size of the spectrum.

3.5.11 Time complexity of the generalised Laplacian matrix

In computing of the generalised Laplacian matrix $\boldsymbol{L}_G$, we need to compute k-path Laplacian matrices for $1 \leq k \leq d_{max}$ which we obtain from the distance matrix $\boldsymbol{S}$ whose elements are the shortest path distances between pairs of nodes in the network. There are various algorithms used in computing the all-pairs shortest paths in a given network as discussed in [23]. The most common one of these algorithms is the Dijkstra's with time complexity $O(mn + n^2 \log n)$ where n and m are the number of nodes and links respectively [32]. Though improvements to Dijkstra's algorithm have been developed [93, 96], for very large networks however, computation of the generalised Laplacian matrix still remains a computationally cumbersome task especially as the size of the network increases. Fortunately, in some applications it is not a requirement to compute all the k-path Laplacian matrices as only the first two or three can be good enough. We however stress that research directed towards developing more efficient algorithms for solving the all-pairs shortest path problem for very large networks would be of utmost importance.

3.5.12 The generalised diffusion model

In modelling the diffusion process in which we consider both direct and long-range interactions, we need to modify the standard diffusion model described by equation 3.2.3 in order to capture the long-range interaction. The modification entails substituting the standard Laplacian matrix $\boldsymbol{L}$ in equation 3.2.3 with the generalised Laplacian matrix $\boldsymbol{L}_G$ (see equation 3.5.6). The generalised diffusion equation is thus given by

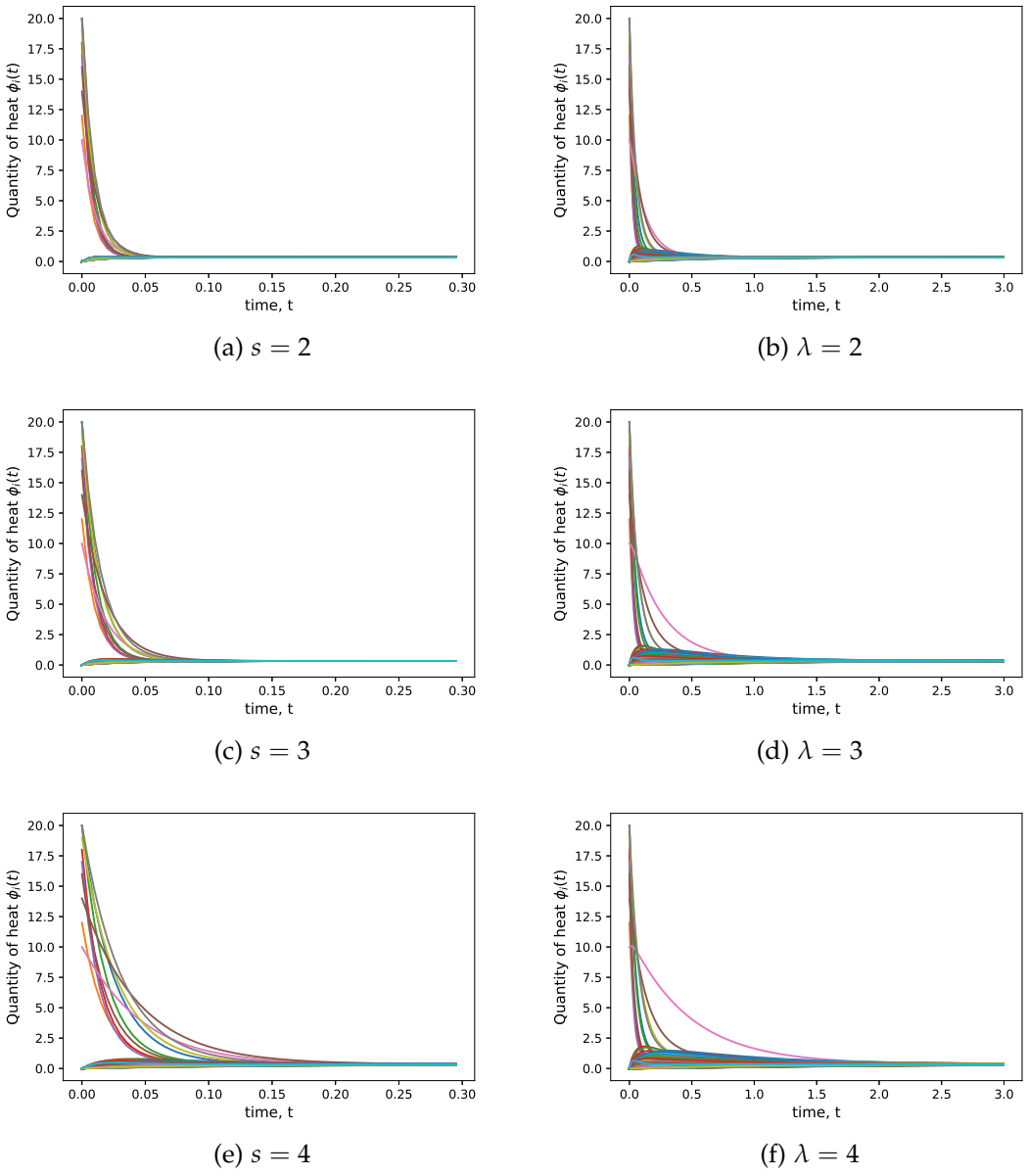
$$\frac{d\boldsymbol{\phi}}{dt} = -\varepsilon \boldsymbol{L}_G \boldsymbol{\phi}, \quad \boldsymbol{\phi}(0) = \boldsymbol{\phi}_0. \tag{3.5.11}$$

From now on, we will take $\varepsilon = 1$ for simplicity.

3.5.13 Comparison of Mellin and Laplace transform-based generalised diffusion

Here, we consider two networks of different structures that is, random networks using Erdős-Rényi (ER) model [35, 68, 88] and power-law networks using Barabási-Albert (BA) model [11, 88] both of size 500 and average degree approximately equal to 4.5. We consider diffusion of heat over both networks. First, for both networks we consider an initial heat distribution in which 10 randomly selected nodes are assigned random values in the interval $[1, 10]$ while the rest of the nodes have values set to zero. We then perform simulations using equation 3.5.11 for both networks for different values of s and λ following the Mellin and Laplace transform-based diffusion approaches as shown in Figure 3.7 and Figure 3.8.

We observe from simulations in Figure 3.7 that generally for both the Mellin and Laplace transform-based cases, the rate of diffusion becomes slower as the values of s and λ increases respectively. For instance at $s = 2$, equilibrium is attained at about 0.06 time steps compared to $s = 3$ where equilibrium is attained at about 0.1 time steps which is slightly double the



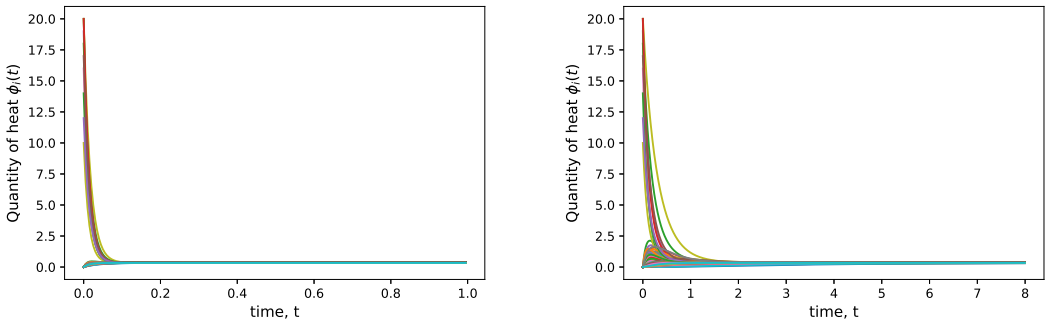
(a) $s = 2$ (b) $\lambda = 2$

(c) $s = 3$ (d) $\lambda = 3$

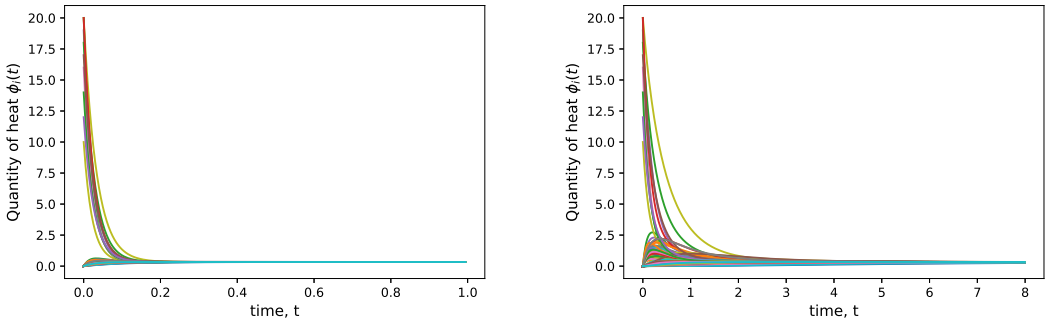
(e) $s = 4$ (f) $\lambda = 4$

Figure 3.7: Results of simulations of diffusion on BA network of 500 nodes and average degree of 4.5. The long-range interactions are accounted for using the Mellin and Laplace transforms of the k-path Laplacian matrices using $s = \lambda = 2$, 3 and 4. The left column corresponds to the Mellin based diffusion while the right column corresponds to the Laplace based diffusion.

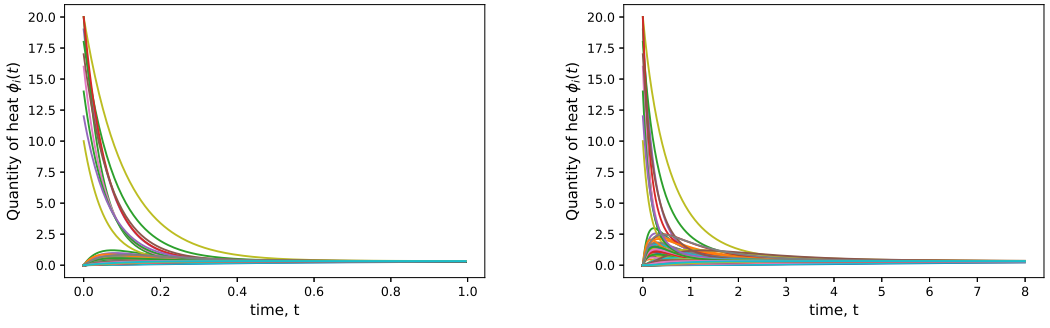
time for the former. Similarly, for the Laplace transform-based case, equilibrium is attained at about $t = 0.8$ and $t = 1.5$ for $\lambda = 2$ and $\lambda = 3$ respectively. This is because as values of s (or λ) increase, the influence of long range interactions weaken and thus a decrease in the rate of diffusion. It is however important to note that on comparing diffusion based on both the Mellin and Laplace transform-based cases, it is evident that rate of diffusion is much faster in the former than later. For instance, we observe that equilibrium is attained at about 0.06 and 0.5 time steps for $s = 2$ and $\lambda = 2$ respectively. This can be explained by the fact that for the same value of exponents s and λ, the corresponding value of strength of long range



(a) $s = 2$ (b) $\lambda = 2$



(c) $s = 3$ (d) $\lambda = 3$



(e) $s = 4$ (f) $\lambda = 4$

Figure 3.8: Simulations (performed using equation 3.5.11) for diffusion on ER network of 500 nodes for which long-range interactions are accounted for using the Mellin and Laplace transforms of the k-path Laplacian matrices using $s = \lambda = 1.5, 2$, and 3. The left column corresponds to the Mellin while the right column corresponds to the Laplace.

interactions is larger for the Mellin than for the Laplace case. When we focus on network with a different structure, that is, the ER network, we observe similar behaviour as for the BA network discussed before. The results of diffusion on ER network for different values of s and λ is illustrated in Figure 3.8.

On comparing diffusion on the BA and ER network for the same values of s and λ, for instance at $s = 3$, equilibrium is attained at about $t = 0.1$ and $t = 0.2$ for BA and ER networks respectively. We can tell that equilibrium state of BA network is reached two times

faster than in the ER network. Turning to the Laplace transform-based case, for instance at $\lambda = 3$, equilibrium is attained at about $t = 1.5$ and $t = 4.0$ for the BA and ER networks respectively. Still like before, equilibrium state is attained about three times faster in BA than in ER network. The faster rate of diffusion in BA networks is attributed to its structure and in particular to the existence of high degree nodes (known as hubs) that disseminate heat faster to the many nodes to connected to them. For the ER networks, however, such hubs (fast spreaders) do not exist and this explains the slow rate at which diffusion occurs as few nodes are reached at a time.

3.5.13.1 Simulations of diffusion on lattice

We consider a 20 by 20 two-dimensional discrete grid in which each point is connected to 8 of its nearest neighbours. Initially, we consider diffusion of heat (represented by a function ϕ) over the graph. We commence by assigning heat quantities of amounts 5 (green), 7 (yellow) and 10 (red) to selected blocks of the lattice and the rest are assigned zeroes (dark blue). We consider two cases of diffusion over the grid namely one through direct interactions of nearest neighbours and the other through both direct and long-range interactions. The latter are accounted for by the Mellin and Laplace transforms of the path Laplacian matrices following equation 3.5.4 and equation 3.5.5 respectively.

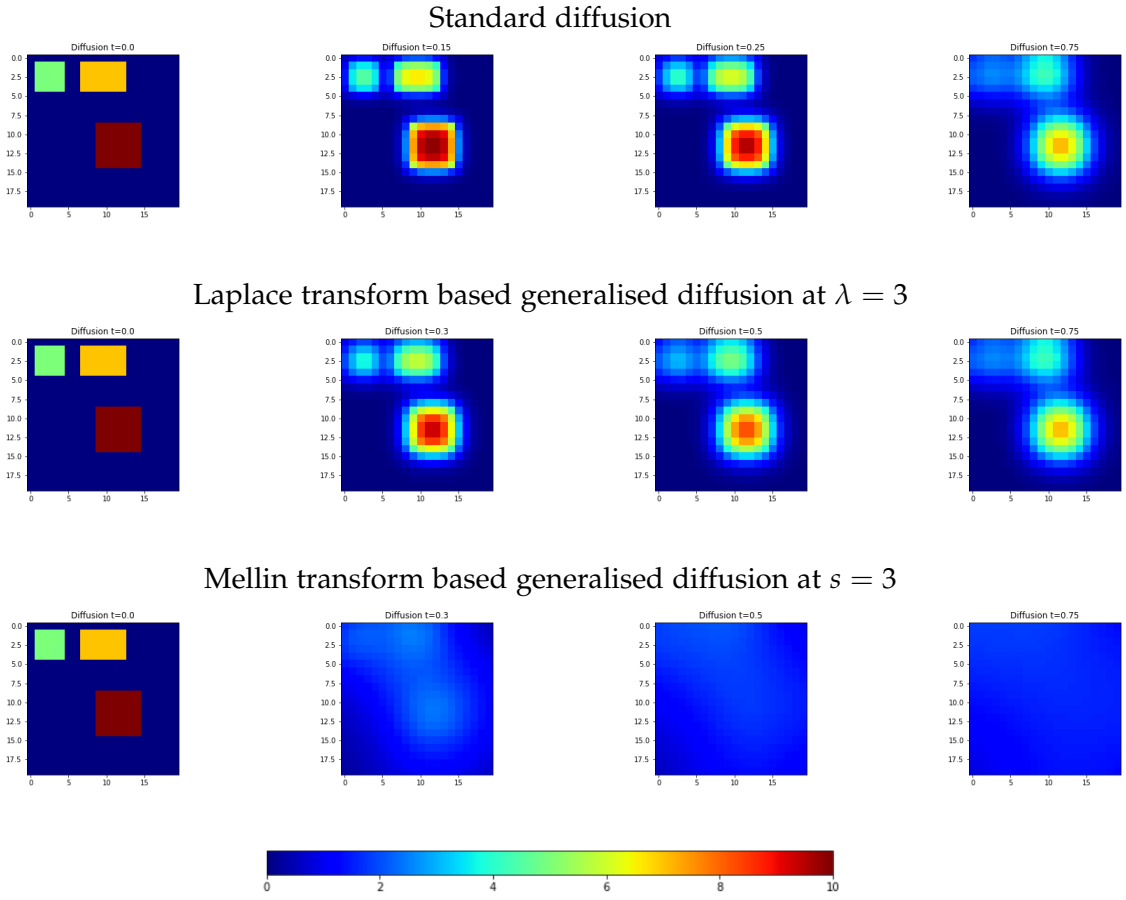


Figure 3.9: Sample illustrations for progression of diffusion over a 20×20 lattice with initial heat quantities indicated by coloured blocks. The diffusion state is captured at different time steps, that is, from left to right, $t = 0$, 2.5, 5.0 and 7.5 respectively for the three cases. The top row corresponds to diffusion through direct interactions only. The middle row and bottom row correspond to diffusion with long-range interactions accounted for by the Laplace and Mellin transforms of k-path Laplacian matrices at $\lambda = s = 3$ respectively.

At the start that is $t = 0$, the state of the lattice is the same in the three cases (direct and long-range cases). However, with time we observe that diffusion occurs much faster in the cases where long-range interactions are involved (middle and bottom rows) compared to the case where diffusion occurs through direct interactions only (top row). Moreover, it is evident from the illustrations in Figure 3.9 that for the case where long-range interactions are accounted for by Mellin transform, diffusion occurs much more faster than the in the Laplace transform-based case. For example, by $t = 7.5$ (column at extreme right) almost the whole lattice is at equilibrium (light blue coloured blocks) for the Mellin transform-based case (bottom row) yet for the Laplace transform-based case (middle row), equilibrium is not yet attained then. Though not indicated in Figure 3.9, for the Laplace-based case and standard diffusion cases, equilibrium is attained at approximately $t = 50$ and $t = 65$ respectively. The observed fast diffusion in the Mellin transform-based case is due to stronger long-range interactions (following from equation 3.5.5) observed in the Mellin transform-based case than in the Laplace transform-based case (given by equation 3.5.4) for the same values of exponents

s and λ respectively. This thus justifies the faster rate of diffusion observed in the former case.

3.6 Conclusion

In this chapter, we have reviewed two models of diffusion on networks. First, the standard model which involves spread of substance along edges of a network from nodes with high quantities of substance to nodes with low quantities of substance until equilibrium state is attained. For a simple undirected network, equilibrium state is attained when all nodes have the same quantity of substance at equilibrium which is the average of the initial quantities at all nodes of the network. Using this model, we further investigated the impact of network structure, choice of source nodes, and heterogeneity of a network to the rate of diffusion. Second, we discuss the generalised diffusion model, introduced by Estrada et al. [41], which is characterised by the spread of substance through both direct interactions along the edges of the network and through long-range interactions between non-nearest nodes. The long-range interactions are accounted for using the Mellin and Laplace transforms of k-path Laplacian matrices [40]. The strength of long-range interactions increases with decrease in respective Mellin or Laplace transform exponents. Furthermore, we performed simulations of diffusion based on the above model on networks of different structures namely the BA and ER networks. The results of simulations show that long-range influence have a stronger impact in the former than the later due to the existence of high degree nodes known as hubs in the former case.

Chapter 4

Heat kernel on graphs

In the previous chapter, we discussed the diffusion process on network in which we presented two models of diffusion. One model, referred to as the standard model, describes the diffusion process in which the flow of substance occurs only along the edges of the network. The other model, which we refer to as the generalised model, captures flow of substance among nearest neighbours via the edges connecting them and also among non-nearest neighbours through long-range interactions as put forward by Estrada [38]. In this chapter, we will focus on the heat kernel which is the fundamental solution to the heat equation. First, we present a review of the heat kernel resulting from the standard diffusion model. This heat kernel has been extensively studied along with its applications in object clustering, centrality measure, community detection, among others [24, 72, 119]. Second, we dive into exploring the heat kernel associated with the generalised diffusion model which we refer to as the generalised heat kernel. We will further discuss various invariants of the generalised heat kernel and ascertain their utility as means of graph characterisation and clustering.

4.1 Introduction

Diffusion on graphs is an important concept in the study and modelling of various real-world systems. As mentioned earlier in chapter 3, the diffusion process in standard and generalised models is captured by equations 3.2.3 and 3.5.11 respectively. In studying the diffusion process on a network, the fundamental solution to the diffusion equation known as the heat kernel, plays a key role and it is perhaps the reason as to why intensive research has been on-going in this area. The heat kernel resulting from the standard diffusion model has been extensively studied along with its applicability to object clustering, centrality measurement, community detection, among others [24, 72, 119]. Particularly in object clustering, a number of invariants of the standard heat kernel have proven to perform well in graph characterisation and object clustering as put forward by authors in [118, 119]. Such invariants include the zeta function, heat kernel trace, derivative of the zeta function at the origin and the heat content.

In [118, 119], it is shown that feature vectors constructed from the above mentioned invariants of the standard heat kernel perform well in clustering of objects using Principal Component Analysis. This prompted us to explore how this story will unveil if we instead consider the same set of invariants of the generalised heat kernel other than the standard one. In this chapter therefore, we will extend the work in [118] by exploring the generalised heat kernel resulting from the generalised diffusion equation (see equation 3.5.11) based on k-path Laplacian matrices proposed in [38, 40]. Moreover, we will ascertain whether the invariants of the generalised heat kernel can also be used for graph characterisation as is the case with the standard heat kernel. Considering the generalised diffusion model based on the Mellin and Laplace transforms of the k-path Laplacian matrices, we will perform experiments on selected objects from the Columbia Object Image Library (COIL-100) [85] database to ascertain the use of the generalised heat kernel invariants for object clustering and the impact of long-range interactions to the quality of object clusters.

4.2 The standard heat kernel

The standard heat kernel, the fundamental solution to the standard diffusion model, has proven very useful in a number of applications such as identification of communities in graph, partitioning of graphs, as a pagerank centrality of a graph, as a means of embedding a graph into a pattern space, among others [24, 25, 72]. As discussed earlier, diffusion of substance along the edges of a graph can be modelled by the equation

$$\frac{d\boldsymbol{\phi}}{dt} = -\boldsymbol{L}\boldsymbol{\phi}, \tag{4.2.1}$$

where $\boldsymbol{L}$ is either the Laplacian matrix or its normalised version.
The standard heat kernel is the fundamental solution to the diffusion equation 4.2.1 and it is obtained by exponentiating the Laplacian eigensystem over time. It is given by

$$\boldsymbol{H}_t = e^{-t\boldsymbol{L}} \tag{4.2.2}$$

It literally describes the flow of substance (for instance heat) across edges (direct interactions) in the graph [119]. On applying spectral decomposition to equation 4.2.2, we have

$$\boldsymbol{H}_t = \boldsymbol{V}e^{-t\boldsymbol{\Lambda}}\boldsymbol{V}^T = \sum_{i=0}^{n} e^{-\lambda_i t}\boldsymbol{v}_i\boldsymbol{v}_i^T \tag{4.2.3}$$

where λ_i denotes the i-th eigenvalue of $\boldsymbol{L}$ with corresponding eigenvector v_i, $\boldsymbol{\Lambda}$ is a diagonal matrix of eigenvalues and $\boldsymbol{V}$ is a matrix with eigenvectors as columns. The eigenvalues are in non-decreasing order $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ (or $0 \leq \lambda_i \leq 2$ for the normalised Laplacian) [7].
For a graph $G = (V, E)$, the standard heat kernel of the graph is an $|V| \times |V|$ matrix whose entry for a pair of nodes p, q is given by

$$\boldsymbol{H}_t(p, q) = \sum_{i=1}^{|V|} e^{-\lambda_i t}\boldsymbol{v}_i(p)\boldsymbol{v}_i(q) \tag{4.2.4}$$

It is noteworthy that $\boldsymbol{H}_t(p,q)$ is the amount of heat flowing from vertex p to vertex q and that $\boldsymbol{H}_t(p,q) \geq 0$ [26].
When t tends to zero, the kernel behaviour can be obtained from the Taylor's expansion of equation 4.2.2 which is

$$e^{-t\boldsymbol{L}} = \sum_{k=0}^{\infty} \frac{(-t)^k}{k!}\boldsymbol{L}^k = \boldsymbol{I} - t\boldsymbol{L} + \frac{t^2}{2!}\boldsymbol{L}^2 - \frac{t^3}{3!}\boldsymbol{L}^3 + \cdots \tag{4.2.5}$$

Thus,

$$\lim_{t \longrightarrow 0}\left(e^{-t\boldsymbol{L}}\right) \approx \boldsymbol{I} - t\boldsymbol{L}. \tag{4.2.6}$$

It therefore implies that for t tending to zero, the heat kernel depends on the local connectivity structure of the graph [118]. On the other hand, as t tends to infinity, following from equation 4.2.3

$$\lim_{t \longrightarrow \infty}\left(e^{-t\boldsymbol{L}}\right) \approx e^{-\lambda_2 t}\boldsymbol{v}_2\boldsymbol{v}_2^T, \tag{4.2.7}$$

where λ_2 is the smallest non-zero eigenvalue and $\boldsymbol{v}_2$ is the corresponding eigenvector (known as Fiedler vector) of the Laplacian matrix. From equation 4.2.7, it is evident that for large t, the heat kernel behaviour is determined by the global structure of the graph [118]. The difference in the behaviour of the heat kernel as t tends to zero and to infinity provides useful insights about the structure of the network and can thus be a useful tool in network analysis.

4.3 The standard heat kernel and path length distribution

In this section, we review the relationship between the standard heat kernel and the distribution of path lengths of a given graph. According to Xiao [118], the derivation of this relationship commences with the relation $\boldsymbol{\mathcal{L}} = \boldsymbol{I} - \boldsymbol{P}$ where $\boldsymbol{\mathcal{L}}$ is the normalised Laplacian matrix and $\boldsymbol{P}$ is the Randić matrix [17]. Using this relation, the heat kernel expression can be re-written as

$$\boldsymbol{H}_t = e^{-t(\boldsymbol{I}-\boldsymbol{P})}. \tag{4.3.1}$$

Using the Maclaurin expansion on the heat kernel, equation 4.3.1 can be expressed as a polynomial in t as

$$\boldsymbol{H}_t = e^{-t}\left(\boldsymbol{I} + t\boldsymbol{P} + \frac{(t\boldsymbol{P})^2}{2!} + \frac{(t\boldsymbol{P})^3}{3!} + \ldots\right) = e^{-t}\sum_{k=0}^{\infty}\boldsymbol{P}^k\frac{t^k}{k!}, \tag{4.3.2}$$

where

$$\boldsymbol{P}^k(p,q) = \sum_{W_k}\prod_{i=1}^{k}\frac{1}{\sqrt{d_{p_i}d_{p_{i+1}}}} \tag{4.3.3}$$

where W_k is a walk of length k with sequence of vertices $p_0, \ldots, p_l$ such that $(p_i, p_{i+1}) \in E$ and d_p is the degree of vertex p. Thus, $\boldsymbol{P}^k(p,q)$ is the sum of weights of all walks of length k connecting nodes p and q. We note that the shortest distance between a given pair of nodes p and q of a graph, G, can be obtained by finding the smallest value of k for which $\boldsymbol{P}^k(p,q)$ is non zero [118], that is,

$$d_G(p,q) = floor_k\boldsymbol{P}^k(p,q). \tag{4.3.4}$$

We recall that the number of walks of length k between a given pair of nodes p and q is equal to the (p,q)-th entry of $\boldsymbol{A}^k$, where $\boldsymbol{A}$ is the adjacency matrix discussed in chapter 2. Using this approach to compute the shortest path between nodes p and q requires inspection through the matrices $\boldsymbol{A}^k$ for different values of k to identify the minimum non-zero (p,q)-th entry. This approach is computationally cumbersome compared to using equation 4.3.4.

Following from equation 4.3.3, the elements of the heat kernel are given by

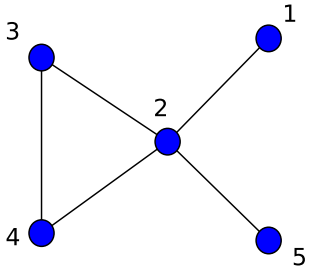
$$\boldsymbol{H}_t(p,q) = e^{-t}\sum_{k=0}^{|V|^2}\boldsymbol{P}^k(p,q)\frac{t^k}{k!} \tag{4.3.5}$$

Since $\boldsymbol{\mathcal{L}} = \boldsymbol{I} - \boldsymbol{P}$, we can deduce that $\boldsymbol{P}^k = (I - \boldsymbol{\mathcal{L}})^k$ and performing eigenvector decomposition of $\boldsymbol{\mathcal{L}}$, we have

$$\boldsymbol{P}^k(p,q) = \sum_{i=1}^{|V|}(1-\lambda_i)^k\boldsymbol{v}_i(p)\boldsymbol{v}_i(q), \tag{4.3.6}$$

where λ_i and $\boldsymbol{v}_i$ are the i-th eigenvalue and corresponding eigenvector of the normalised Laplacian matrix $\boldsymbol{\mathcal{L}}$.

Example 4.3.1. Let us consider a simple graph in Figure 4.1 with its normalised Laplacian matrix, $\boldsymbol{\mathcal{L}}$, as shown below:



$$\boldsymbol{\mathcal{L}} = \begin{pmatrix} 1 & -0.5 & 0 & 0 & 0 \\ -0.5 & 1 & -0.354 & -0.354 & -0.5 \\ 0 & -0.354 & 1 & -0.5 & 0 \\ 0 & -0.354 & -0.5 & 1 & 0 \\ 0 & -0.5 & 0 & 0 & 1 \end{pmatrix}$$

Figure 4.1: A simple graph of size 5 and its normalised Laplacian matrix.

We then compute the shortest path distance between node 1 and node 4 using equation 4.3.4 in which $\boldsymbol{P}^k$ is computed using equation 4.3.6. We obtain $\boldsymbol{P}^1(1,4) = 0$, $\boldsymbol{P}^2(1,4) = 0.1768$, $\boldsymbol{P}^3(1,4) = 0.0884$, and $\boldsymbol{P}^4(1,4) = 0.1768$. Thus, for non-zero $\boldsymbol{P}^k$, $floor_k\boldsymbol{P}^k(1,4) = 2$ which implies that the shortest distance $d_{1,4} = 2$ which is the path $p(1,2,4)$.

4.4 The trace of the heat kernel

The combinatorial trace of the heat kernel at time t, denoted as $Tr(\boldsymbol{H}_t)$, is the sum of the entries at the main diagonal of the heat kernel matrix. It is given by

$$Tr(\boldsymbol{H}_t) = Tr(\boldsymbol{V}e^{-t\boldsymbol{\Lambda}}\boldsymbol{V}^T) = Tr(e^{-t\boldsymbol{\Lambda}}(\boldsymbol{V}^T\boldsymbol{V})) = Tr(e^{-t\boldsymbol{\Lambda}}), \tag{4.4.1}$$

where $\boldsymbol{V}$ is the matrix with eigenvectors of $\boldsymbol{L}$ as columns and $\boldsymbol{\Lambda}$ is the diagonal matrix with eigenvalues λ_i. Thus, the trace of the heat kernel is a function, $Z(t)$, whose parameters are the eigenvalues of the Laplacian matrix and whose argument is time. It is given by

$$Z(t) = Tr(\boldsymbol{H}_t) = \sum_p \boldsymbol{H}_t(p,p) = \sum_i e^{-\lambda_i t}, \tag{4.4.2}$$

where λ_i is the eigenvalue of the Laplacian matrix [118]. From equation 4.4.2, it is evident that the trace of the heat kernel is invariant to permutations. For a connected graph, equation 4.4.2 can be written as

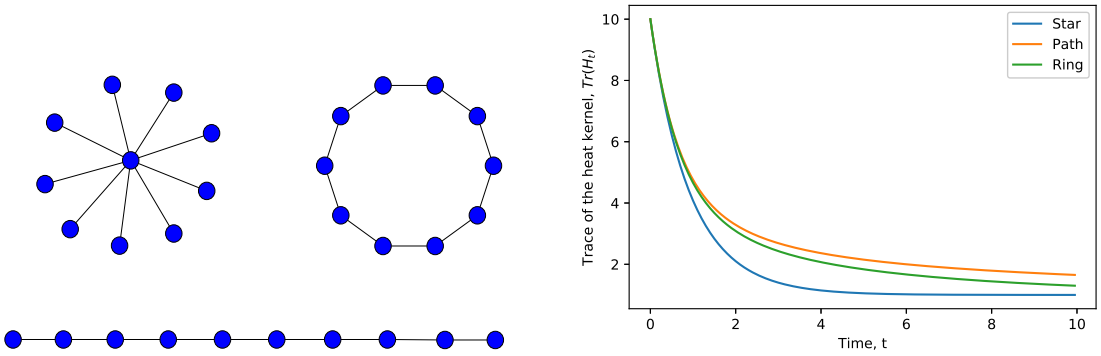
$$Z(t) = 1 + e^{-\lambda_2 t} + e^{-\lambda_3 t} + \ldots + e^{-\lambda_n t}, \tag{4.4.3}$$

where n is the number of vertices of a graph.

In most literature, the trace of the heat kernel is used rather than the heat kernel due to the fact that the trace can capture the essential part of the heat kernel and can be computed in polynomial time (following from equation 4.4.2). In other words, the trace helps to overcome the cumbersome task associated with computing the heat kernel and it is thus an effective tool that can capture graph properties as well as major invariants. For example, for a complete graph, K_4, for all vertices, p, the entry $\boldsymbol{H}_t(p,p)$ is the same which makes computation of the trace much easier.

4.4.1 Graph characterisation using the heat kernel trace

According to Xiao [118], the trace of the heat kernel has a potential applicability to distinguishing graphs with different topologies based on the shape of the curves obtained by plots of the trace of the heat kernel as a function of time. For illustration of this concept, we consider 3 simple graphs namely a star, path and 2-regular graph of size 10 each. Figure 4.2 shows the plot of heat kernel trace against time for the three graphs.



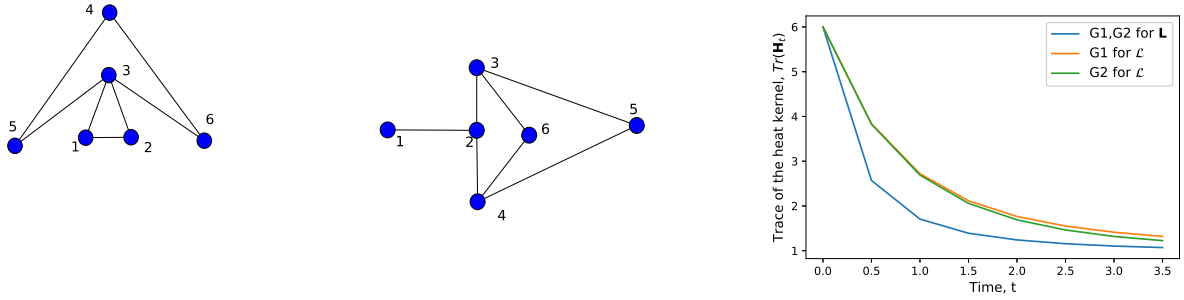
(a) (b)

Figure 4.2: (a) three graphs used for which analysis is performed. (b) plot of the heat kernel trace against time for star (blue), path (orange) and regular (green) graphs.

From Figure 4.2, we observe that since the 3 graphs have different topologies, the corresponding curves take on different shapes as well, that is to say, the curves are distinct. It is evident

that the curves corresponding to the path graph and 2-regular graph (ring) graph are closer to each other as both graphs have almost similar topologies. On the other hand, the curve corresponding to the star graph is quite separate and has a different (deeper trough) shape.

Based on the study we carried out, the use of the heat kernel trace for graph characterisation has limitations. One of them is the difficulty in characterising co-spectral non-isomorphic graphs. These are graphs which are not isomorphic but have the same multi-set of eigenvalues. Plots of the trace function against time for such graphs do coincide due to similarity in spectrum and thus these graphs may be characterised as structurally similar which is not true. An example of co-spectral non-isomorphic graphs is shown in Figure 4.3 along with the corresponding plot of the trace function of the heat kernel.



(a) (b) (c)

Figure 4.3: (c) Plot of the trace function of the normal Laplacian matrix against time for two co-spectral graphs (a) and (b) with respect to $\boldsymbol{L}$.

From Figure 4.3c, it is evident that curves of the trace plot based on the Laplacian matrix against time for the two graphs coincide due to a similar multi-set of eigenvalues of the Laplacian matrix shared by both graphs. However, since the graphs are not co-spectral with respect to the normalised Laplacian matrix, we observe distinct curves of the trace function for the two graphs. Based on this evidence, we can thus report that the heat kernel trace cannot be used to characterise co-spectral non-isomorphic graphs, as they display similarity in structure following from the plots of the trace function and yet their structures are quite different.

In addition, Xiao [118] highlighted another limitation that is attributed to the fact that for each value of time, t, only a single scalar attribute is provided. This calls for either sampling of the trace function with time or selecting a fixed value of time to work with.

4.5 The generalised heat kernel

In the previous section, we have discussed the heat kernel resulting from the diffusion process in which the spread occurs through direct interactions, that is to say, along edges connecting nearest neighbour nodes in the network. In this section, we extend this concept by considering the heat kernel resulting from the generalised diffusion equation (equation 3.5.11), that is to say, where long-range interactions are taken into account. We refer to this heat kernel as the generalised heat kernel and it is given by

$$\boldsymbol{H}_{G_t} = e^{-t\boldsymbol{L}_G}, \tag{4.5.1}$$

where

$$\boldsymbol{L}_G = \Big(\sum_{k=1}^{\Delta} c_k \boldsymbol{L}_k \Big) = c_1\boldsymbol{L}_1 + c_2\boldsymbol{L}_2 + \ldots + c_\Delta \boldsymbol{L}_\Delta, \tag{4.5.2}$$

where $k \in \mathbb{N}$, $1 \leq \Delta \leq d_{max}$ (diameter of network) and c_k are the coefficients. As discussed earlier, the coefficients c_k are chosen in such a way that as distance k increases, the long range effect is weakened. Some of the common expressions for coefficients c_k are $c_1 = 1, c_{k\geq 2} = e^{-\lambda k}$ or $c_{k\geq 2} = k^{-s}$ which depict physical influence [38]. Equation 4.5.2 can be written as

$$\boldsymbol{H}_{G_t} = e^{-t(c_1\boldsymbol{L}_1 + c_2\boldsymbol{L}_2 + \ldots + c_\Delta \boldsymbol{L}_\Delta)}, \tag{4.5.3}$$

where $\boldsymbol{L}_1, \boldsymbol{L}_2, \ldots, \boldsymbol{L}_\Delta$ are k-path Laplacian matrices for hops of length $k = 1, 2, \ldots, \Delta$ respectively. We note that at $k = 1$, we recover the standard heat kernel in equation 4.2.2.

The (generalised) Laplacian matrix plays a key role in determining the (generalised) heat kernel. On performing permutations of node labels of a graph, the spectrum of (generalised) Laplacian matrix of the graph remains unchanged. Thus, functions whose arguments are the spectrum of the (generalised) Laplacian are considered invariants under node label permutations. Xiao [118] presented an extensive study of some of the invariants of the standard heat kernel. These include the trace of the heat kernel, zeta function, derivative of the zeta function at the origin and the heat content. In subsequent sections, we will explore a similar set of invariants associated with the generalised heat kernel.

4.6 Trace of the generalised heat kernel

Like the trace of the standard Laplacian matrix discussed earlier on, we define the trace of the generalised heat kernel by

$$Z_G(t) = Tr(\boldsymbol{H}_{G_t}) = \sum_{i=1}^{|V|} e^{-\mu_i t}, \tag{4.6.1}$$

where μ_i is the i-th eigenvalue of the generalised Laplacian matrix. Alternatively, for the same graph with n vertices, equation 4.6.1 can be written as

$$Z_G(t) = 1 + e^{-\mu_2 t} + e^{-\mu_3 t} + \ldots + e^{-\mu_n t}, \tag{4.6.2}$$

which takes on a similar format as equation 4.4.3 for the trace of the standard Laplacian matrix.

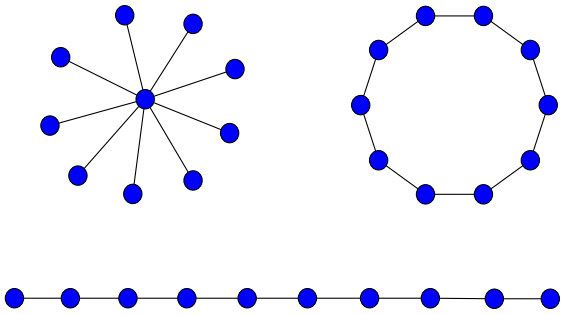
We earlier on pointed out that the multiplicity of zero as an eigenvalue of $\boldsymbol{L}_G$ is equal to the number of connected components in a given graph; it therefore follows from equation 4.6.2 that the trace of the generalised heat kernel can be re-written as

$$Tr(\boldsymbol{H}_{G_t}) = C_G + \sum_{\mu_i \neq 0} e^{-\mu_i t}, \tag{4.6.3}$$

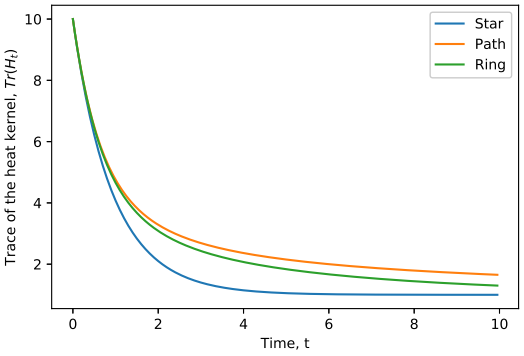
where C_G is the multiplicity of zero as an eigenvalue of $\boldsymbol{L}_G$, that is to say, the number of connected components of a graph.

We are interested in ascertaining whether the trace of the generalised heat kernel can be used as a basis for analysing graphs as is the case with the trace of the standard heat kernel discussed earlier on. We consider a simple example of three graphs namely the star, ring and path graphs of size 10 each as shown in Figure 4.4a. Using the generalised Laplacian matrix based on both the Mellin and Laplace transforms of the k-path Laplacian matrices, we perform simulations and results of plots of the trace function against time, t, shown in Figure 4.4.

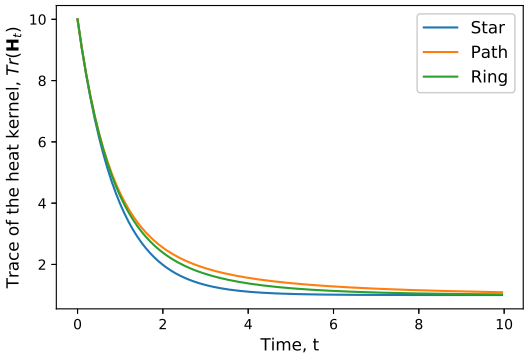
From Figure 4.4, we observe distinct curves with distinct shapes for the 3 graphs for the trace of the heat kernel of both the standard and generalised heat kernel. Focusing on the middle row which corresponds to the Mellin transform-based generalised trace function, we observe that for $s = 2$ (Figure 4.4c), the curves get much closer to each other, nevertheless, we can still observe their distinctiveness, albeit their slopes are slightly different from each other. For $s = 3$ (Figure 4.4d), the curves corresponding to the 3 graphs are more distinct and their shape tends to the one of the trace plots of the standard heat kernel (see Figure 4.4b). This can be attributed to a decrease in strength of long-range interactions as s increases. For the Laplace transform-based case illustrated in the bottom row (see Figures 4.4e and 4.4f), we observe a similar behaviour as in the Mellin case. A slight difference between the two cases is that as s and λ increase, the trace plots in the Laplace transform-based case become similar to the trace plots of the normal Laplacian matrix at a faster rate than in the Mellin transform-based case. This is due to stronger long-range influence in the Mellin transform-based case than the Laplace transform-based case for the same value of the respective exponents. Thus, as the values of the exponents increase, there is a decrease in long-range influence with a rate much faster in the Laplace than the Mellin transform-based case. Based on the results of the simulations in Figure 4.4, we can thus conclude that the trace function of the generalised heat kernel can as well be used as a tool for analysing graphs with different topologies.



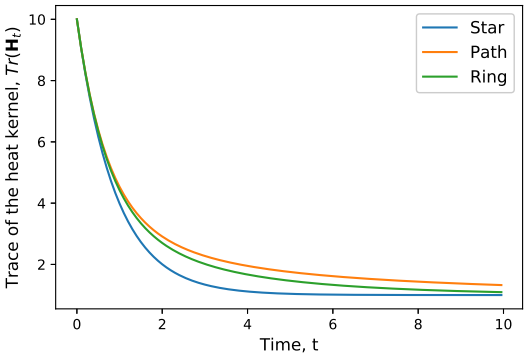
(a)



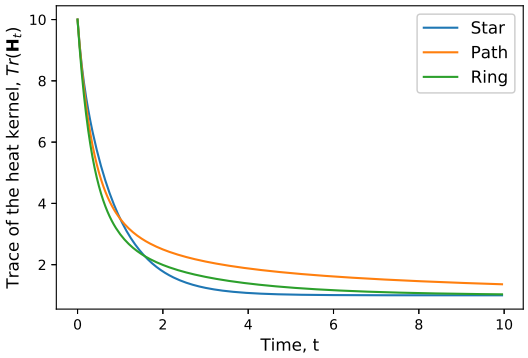
(b)



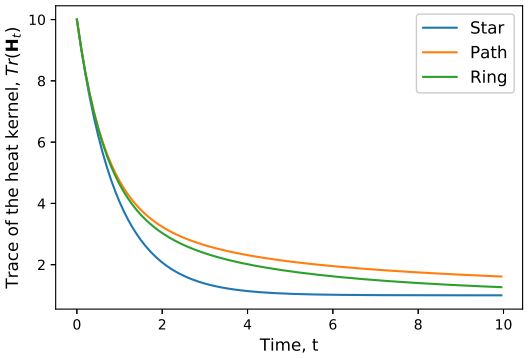
(c) $s = 2$



(d) $s = 3$



(e) $\lambda = 2$



(f) $\lambda = 3$

Figure 4.4: Plots of the trace of the generalised heat kernel against time for the star (blue curve), ring (green curve) and path (orange curve) graphs in (a). (b) is the plot of the trace of the standard heat kernel. (c) and (d) in the middle row correspond to plots of the trace function for the generalised heat kernel using Mellin transform at $s = 2$ and $s = 3$ respectively. The bottom row, that is, (e) and (f) are plots of trace function of the generalised heat kernel based on Laplace transform at $\lambda = 2$ and $\lambda = 3$ respectively.

Let us consider a simple toy example to illustrate the variation of the trace of the generalised heat kernel with time for different values of Mellin exponent s and Laplace exponent λ.

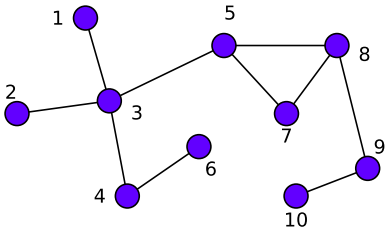
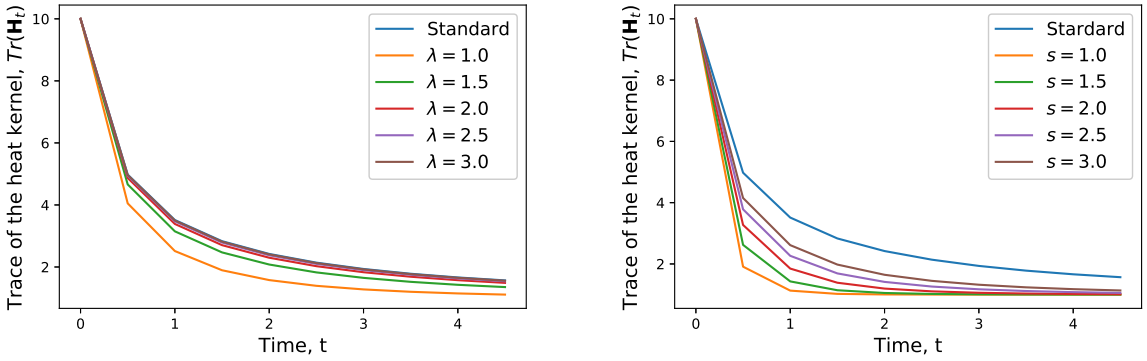


Figure 4.5: A simple network of size 10



(a) (b)

Figure 4.6: Plots (performed using equation 4.6.1) of the trace of the generalised heat kernel against time for the simple graph in Figure 4.5. The long-range interactions are accounted for by the Laplace (left) and Mellin (right) transforms of the k-path Laplacian matrices of the graph for different values of λ and s respectively.

First, we observe from the plots in Figure 4.6 that the curve for the trace function against time for the standard diffusion process (that is, through direct interactions) on the graph (in blue) is the top most curve and it decreases gradually with time. When we consider long-range influence, we observe that as the values of the parameters λ and s for the Laplace and Mellin transforms respectively increase, the corresponding curves approach that of the standard diffusion (in blue). However, we note that the rate of tendency to the standard diffusion curve is faster in the Laplace transform-based case than in the Mellin transform-based case. This is due to the fact that long-range influence is stronger in the Mellin case than in the Laplace case for the same value of respective exponents following from equations 3.5.5 and 3.5.4. For clarity, let us consider $\lambda = s = 3.0$ (brown); we observe from Figure 4.6 that the curve corresponding to Laplace transform-based case already coincides with that of the standard diffusion curve while that for the Mellin transform-based case is still distant from the standard diffusion curve.

4.7 The zeta function

In the literature, there exist various definitions of the zeta function for finite simple graphs [47, 73]. In this thesis, we focus on the zeta function associated with the eigenvalues of the Laplacian matrix which is obtained by exponentiating and summing the reciprocal of the non-zero eigenvalues [47]. For the standard Laplacian matrix, $\boldsymbol{L}$, the associated zeta function is defined by

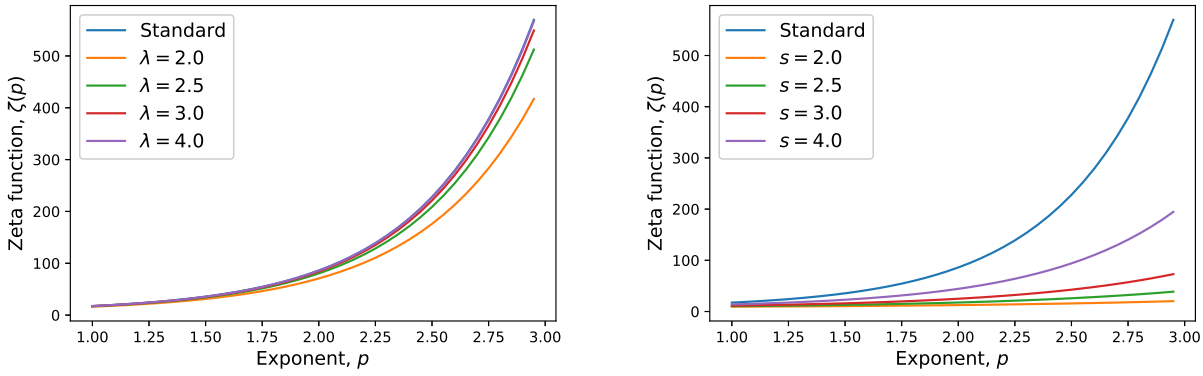
$$\varsigma(p) = \sum_{\lambda_i \neq 0} \lambda_i^{-p}, \tag{4.7.1}$$

where λ_i is the i-th eigenvalue of the standard Laplacian matrix. Following from the above definition, we define the zeta function associated with the generalised Laplacian matrix as

$$\zeta(p) = \sum_{\mu_i \neq 0} \mu_i^{-p}, \tag{4.7.2}$$

where μ_i denotes the i-th eigenvalue of the generalised Laplacian matrix.

Let us consider a plot of the zeta function against p for different values of the Mellin transform and Laplace transform exponents s and λ respectively for a simple graph of size 10 (see Figure 4.5).



(a) (b)

Figure 4.7: Illustration of the zeta function of the graph in Figure 4.5 against exponent p. (a) corresponds to the Laplace transform-based case with $\lambda = 2, 2.5, 3$, and 4. (b) corresponds to the Mellin transform-based case with $s = 2, 2.5, 3$, and 4.

For the standard Laplacian and the Mellin or Laplace transform-based generalised Laplacian, we observe that the zeta function increases with increase in the exponent p. For different values of Laplace exponent λ, we can tell from the illustration in Figure 4.7a that the variation of the zeta function with p follows a similar trend to that in the standard Laplacian curve (in blue) due to relatively less pronounced long-range influence which also decreases with increase in the value of λ. On the contrary, as the Mellin exponent, s, changes, there are observable changes in the corresponding curves for the zeta function against p (see Figure 4.7b).

This can be attributed to stronger long-range interactions in the Mellin transform-based case than in the Laplacian transform-based case. Like in the former case, the strength of long-range influence decreases as the exponent s increases which explains why the curve corresponding to $s = 4$ (purple) gets closer to one of the standard Laplacian matrix (blue).

4.7.1 The zeta function and generalised heat kernel trace moments

In his work [118], Xiao showed that the zeta function and the heat kernel trace, both associated with the standard Laplacian matrix, are related in some way using the Mellin transform. In a similar way, we explore this relationship for the case of the generalised heat kernel and the zeta function of the eigenvalues of the generalised Laplacian matrix.
Let us consider the function $f(t) = e^{-\mu_i t}$. Its Mellin transform is given by

$$\mu_i^{-p} = \frac{1}{\Gamma(p)} \int_0^\infty t^{p-1} e^{-\mu_i t} dt, \tag{4.7.3}$$

where μ_i is the i-th eigenvalue of $\boldsymbol{L}_G$ and $\Gamma(p)$ is the gamma function defined by

$$\Gamma(p) = \int_0^\infty t^{p-1} e^{-t} dt. \tag{4.7.4}$$

On summation of all non-zero eigenvalues of the Laplacian, equation 4.7.3 becomes

$$\zeta(p) = \sum_{\mu_i \neq 0} \mu_i^{-p} = \frac{1}{\Gamma(p)} \int_0^\infty t^{p-1} \sum_{\mu_i \neq 0} e^{-\mu_i t} dt \tag{4.7.5}$$

Using the connected component based formula for the trace of the generalised heat kernel (see equation 4.6.3), equation 4.7.5 becomes

$$\zeta(p) = \frac{1}{\Gamma(p)} \int_0^\infty t^{p-1} \left\{ Tr(\boldsymbol{H}_{G_t}) - C_G \right\} dt, \tag{4.7.6}$$

where C is the number of connected components of the graph. Thus the zeta function is related to the moments of the heat kernel trace.

4.8 Derivative of the zeta function at the origin

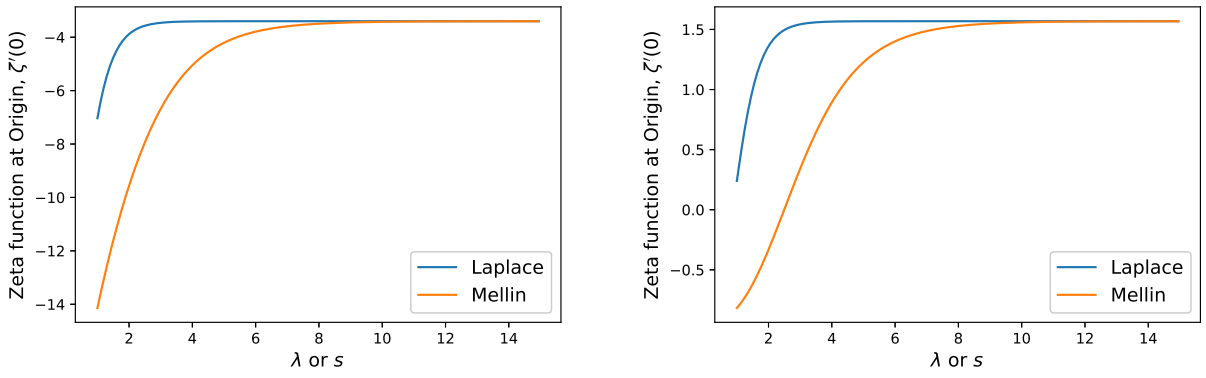
The derivative or slope of the zeta function at the origin is another invariant of the heat kernel that can be used for graph characterisation [119]. In this thesis, we focus on the derivative of the zeta function at the origin associated with the generalised heat kernel with intention of ascertaining its utility for graph characterisation. To begin with, the derivative of the zeta function associated with the generalised Laplacian matrix is given by

$$\zeta'(p) = \sum_{\mu_i \neq 0} \{-\ln \mu_i\} e^{-p \ln \mu_i}. \tag{4.8.1}$$

At the origin, that is $p = 0$, we have

$$\zeta'(0) = -\sum_{\mu_i \neq 0} \ln \mu_i \tag{4.8.2}$$

This is another simple characterisation of graphs. It is simpler than the zeta function because it is parameter independent.



(a) (b)

Figure 4.8: Derivative of the zeta function at origin against time for the graph in Figure 4.5. Panel (a) corresponds to the generalised Laplacian matrix, $\boldsymbol{L}_G$, while the plot in panel (b) corresponds to the generalised normalised Laplacian $\boldsymbol{\mathcal{L}}_G$.

To start with, we compute the values of the derivative of the zeta function at the origin for the graph (Figure 4.5) which are 1.5686 and -3.4012 for the normalised and unnormalised Laplacian matrices respectively. We then turn our attention to the generalised version of both the normalised and unnormalised Laplacian: From Figure 4.8, we observe that in both cases, the values of $\zeta'(0)$ increase rapidly as the Mellin (or Laplace) exponents, s (or λ), increase until constant values of $\zeta'(0)$ are attained at s (or λ) ≈ 9. The observed behaviour is due to the fact that as the values of s (or λ) increase, the strength of long-range interactions decreases until a point in which further increase in the value of the exponents results in negligible strength of long-range interactions and in turn a corresponding negligible change in the value of the spectrum of the corresponding generalised Laplacian matrix. We, however, observe that for the Laplace transform-based case, the value for the derivative of zeta function at the origin approaches faster (in both (4.8a) and (4.8b)) that of the standard diffusion (no long range interactions) compared to the Mellin transform-based case. This is due to a smaller value of long-range interactions in the former than in the latter case following from equations 3.5.4 and 3.5.5 which are used to compute the strength of interactions in the two cases respectively.

4.9 The generalised heat content

The generalised heat content is defined as the sum of entries of the generalised heat kernel matrix (see equation 4.5.2) of a graph. For a graph $G = (V, E)$, the generalised heat content, denoted by Q_t, at time, t, is given by

$$Q(t) = \sum_{p \in V} \sum_{q \in V} \boldsymbol{H}_{G_t}(p, q) \tag{4.9.1}$$

We note that, when we consider a particular case of hops of length 1 ($k = 1$), the corresponding formulation of equation 4.9.1 is equal to the heat content based on the standard Laplacian matrix which is extensively presented in [118].
Following from equation 4.9.1, the heat content for a given pair of nodes p and q is given by

$$Q(t) = \sum_{p \in V} \sum_{q \in V} \sum_{i=1}^{|V|} e^{(-\mu_i t)} v_i(p) v_i(q). \tag{4.9.2}$$

According to the authors in [81], the heat content can be expanded into a polynomial in time and it is given by

$$Q(t) = \sum_{m=0}^{\infty} q_m t^m, \tag{4.9.3}$$

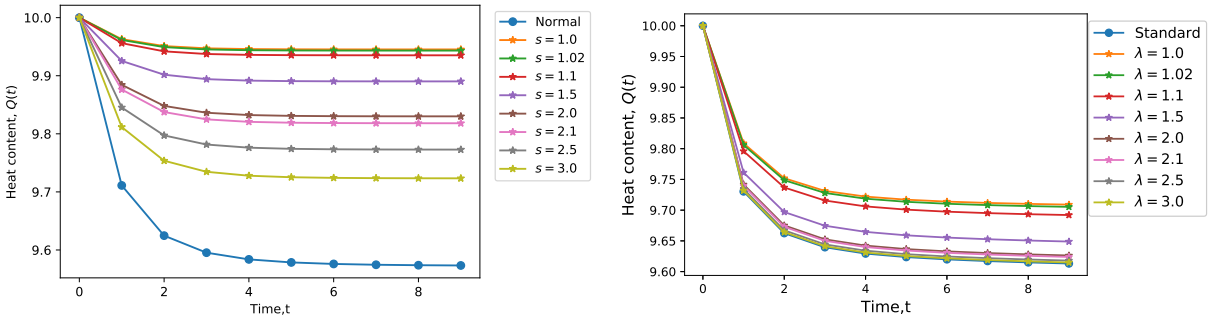
where q_m is given by

$$q_m = \sum_{i=1}^{|V|} \left\{ \left(\sum_{p \in V} v_i(p) \right)^2 \right\} \frac{(-\mu_i)^m}{m} \tag{4.9.4}$$

From equation 4.9.4, we can tell that any coefficients, q_m, can be computed using the entries of eigenvectors and eigenvalues and thus the set of polynomial coefficients, q_m, is unique for a given graph. This property can be used as a basis for graph characterisation. For purposes of graph clustering, we can construct feature vector using the k-leading coefficients, that is, $B_k = (q_1, q_2, \ldots, q_k)^T$. We will further explore this concept in section 4.11.

4.9.1 The generalised heat content simulations

As pointed out before, the normalised Laplacian matrix performs better than the normal Laplacian in some scenarios. For computation of heat content, we use the normalised Laplacian since for the unnormalised case, the heat content remains constant over time.



(a) (b)

Figure 4.9: Results of simulations of the generalised heat content against time for simple graph of size 10 and average degree equal to 2.0 (see Figure 4.5). Panels (a) and (b) correspond to the results for different values of the parameters, s and λ of the Mellin and Laplace transform-based generalised normalised Laplacian matrix respectively.

At $t = 0$, the heat kernel matrix $\boldsymbol{H}_{G_0} = \boldsymbol{I}$. Thus, the value of heat content of a given graph at $t = 0$ is equal to the trace of $\boldsymbol{I}$ which is equal to the number of vertices, $|V|$, of the graph. For a simple graph of size 10 (Figure 4.5), we have $Q(0) = 10$.

In Figure 4.9, the curve (in blue) corresponds to diffusion that occurs only along the edges of the graph, we observe that the heat content decreases rapidly with time. However, on comparing with the generalised diffusion based on both Laplacian and Mellin transform-based cases, we can see that as the respective values of λ and s increase, there is a faster drop in heat content and the heat content curves tend to the one corresponding the standard diffusion process in both cases. It is, however, evident that the rate at which the heat content drops is faster in the Laplace transform-based case than in the Mellin transform-based case which is due to a stronger influence in long-range interactions compared to the Laplace transform-based case which is attributed to bigger values of coefficients c_k in the former case (equation 3.5.5) than in the latter (equation 3.5.4) for same values of k and exponents s and λ respectively. Following from equation 4.9.5, as time tends to infinity, the heat content tends to zero.
It is worth noting that for very large graphs, computation overheads for the heat content are very high. To overcome this problem, Kang [67] put forward methods to estimate heat content based on matrix multiplication and random walks.

4.9.2 Asymptotic behaviour of the heat content

According to Kang [67], equation 4.9.2 can be written as

$$Q(t) = \sum_{i=1}^{|V|} \alpha_i e^{-\mu_i t}, \tag{4.9.5}$$

where $\alpha_i = \sum_{p \in V} \sum_{q \in V} v_i(p) v_i(q)$. We can see from equation 4.9.5 that the heat content can be treated as a summation of exponential functions with different decay rates determined by Laplacian eigenvalues and different weights (α_i) determined by the eigenvectors of the Laplacian. The asymptotic behaviour near zero of the normalised heat content (obtained by dividing through by the largest value at each time, t) can be described as, $Q_n(t)|_{t \longrightarrow 0^+} = \sum_i \alpha_i = 1$. For a given initial distribution of heat on a graph, the heat content describes the diffusion of heat with time on the graph. The asymptotic behaviour of the heat content has been used in studying the structure of graphs [67].

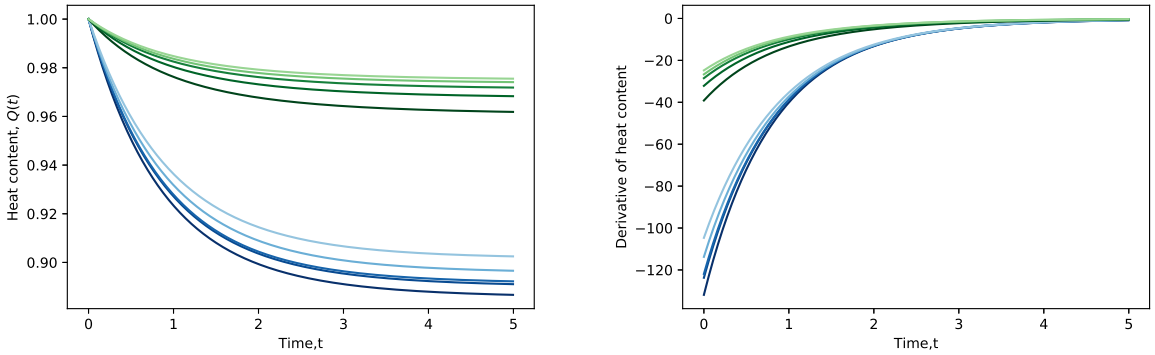
From equation 4.9.5, we can tell that the trivial eigenvalue ($\mu_i = 0$) has a large corresponding weight value, α. In addition, the exponential decay component associated with the trivial eigenvalue drops much slower and thus dominates the heat content curve. In order to do away with the dominance of the trivial eigenvalue and to ascertain the role played by large eigenvalues, Kang [67] put forward the idea of considering first and second time derivatives of the heat content given by the equations below:

$$\frac{\partial Q}{\partial t} = -\sum_{i=1}^{|V|} \alpha_i \mu_i e^{-\mu_i t} \qquad \text{and} \qquad \frac{\partial^2 Q}{\partial t^2} = \sum_{i=1}^{|V|} \alpha_i \mu_i^2 e^{-\mu_i t} \tag{4.9.6}$$

Following from the above derivative equations, smaller eigenvalues have smaller corresponding weights while larger eigenvalues have larger weights associated with them. Moreover, the asymptotic behaviour of the first and second derivatives of the heat content are given by

$$\frac{\partial Q}{\partial t}|_{t\longrightarrow 0^+} = -\sum_i \alpha_i \mu_i \qquad \text{and} \qquad \frac{\partial^2 Q}{\partial t^2}|_{t\longrightarrow 0^+} = \sum_i \alpha_i \mu_i^2. \tag{4.9.7}$$

Thus, from the asymptotic behaviour one can easily distinguish graphs of different structures quickly using the initial time derivatives of the heat content. Based on this property, Kang [67] developed a fast algorithm for differentiating graphs which he used to compare random graphs generated by the ER model and graphs with heavy tail degree distribution generated by the BA model. From the results of the simulations, the heat content curves of the latter dropped much faster than that for the former due to the different α values associated with the eigenvalues of the Laplacian except the smallest eigenvalue. Furthermore, in comparison with other graph similarity testing algorithms, Kang pointed out the following advantages of his heat content-based algorithm: First, a clear difference of the two graphs was evident at the very beginning of the heat content curves. Second, the algorithm does not require prior nodes correspondence. Finally, the algorithm is robust to minor changes in comparatively large graphs and can thus be used in random complex networks. Interestingly, for graphs with heavy tail distributions, the asymptotic behaviour of the heat content can distinguish graphs with different average degrees as shown in Figure 4.10b where the curves with deep blue colours are ones of lower average degree and the lighter ones are ones with higher average degree.



(a) (b)

Figure 4.10: (a) is a plot of normalised heat content $Q_n(t)$ with time for five BA (blue lines) and five ER graphs (green lines) made of 1000 nodes each with average degree between 5 and 20. (b) is the plot of the first derivative of the heat content with time.

From Figure 4.10, it is evident that as $t \rightarrow 0^+$, it is very difficult to distinguish graphs. Particularly at $t = 0$, all graphs have the same value of heat content which is equal to 1 (see Figure 4.10a). However, when we consider the plot of the first derivative of the heat content (see Figure 4.10b), we observe that different graphs can be distinguished immediately as $t \rightarrow 0^+$.

4.10 Graph characterisation using heat kernel invariants

4.10.1 Image representation using Delaunay graphs

Graph-based techniques are widely used in a number of applications such as computer vision, image processing and analysis, pattern recognition, object clustering, among others [66]. In high level vision for example, graph structures have been used to represent structural and relational arrangements of objects in a scene [66]. This is achieved through a graph representation where nodes represent the objects or parts of objects, while the edges (or links) describe relations between the objects or parts of the objects respectively. The idea behind graph-based techniques is to interpret the concept of interest as a graph theory concept to which methods and theories such as spectral graph theory, graph-cuts theory, random walks and graph similarity can be applied. For example object classification can be viewed as a graph clustering problem when different objects are represented by graphs. Kang [67] motivated the effectiveness of graph-based methods for image representation based on the fact that in the human brain, information is stored in graph form where nodes represent neurons and links represent connections between neurons and this arrangement could perhaps be responsible for the speed at which the human brain stores and retrieves information. Further more, Shokoufandeh et al. [99] showed the utility of graph structures in structural abstractions of 2D and 3D objects in object recognition and learning of shape classes.

There are different techniques of representing objects using graphs. The nature of the application at hand dictates the technique used. For example, in image segmentation problems, pixel-based method is often used. In this method, every pixel in an image is represented by a vertex while weighted edges are connections between neighboring pixels based on intensity differences [50]. After extracting the graph, the image segmentation problem can then be solved using graph theory concepts such as random walks [50], graph cuts especially minimum-cut and maximum-flow methods [97], among others. Despite its rampant usage, one drawback of the pixel-based method is that when the resulting graph is large, it becomes challenging to apply techniques such as spectral graph theory as computation overheads are high. In order to overcome this problem, improvements in this method have been developed such as the use of super-pixels (that is to say, groups of very similar pixels with almost the same color and intensity contents) rather than individual pixels as vertices of a graph [1]. Another approach is discussed in [56] where an image is represented by an undirected labelled planar graph known as a segmentation graph whose vertices are singly-connected regions

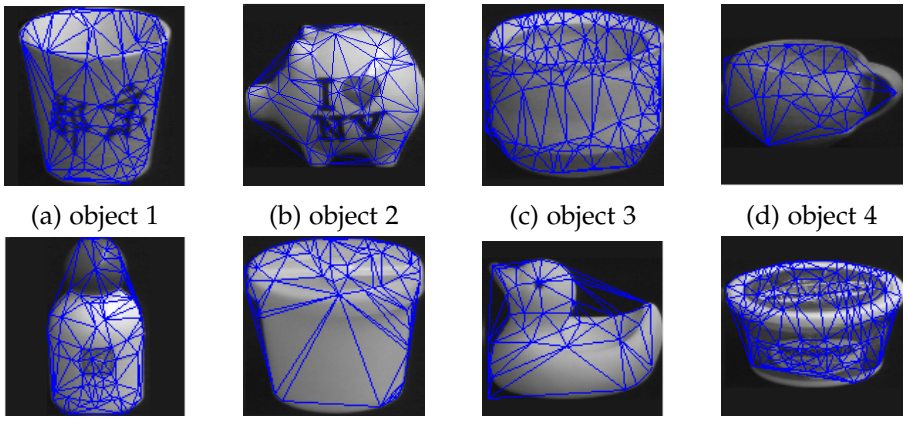
with edges connecting neighbouring regions.

Another approach involves a combination of corner point detection (specifically the Harris corner detection) and Delaunay triangulation techniques. The process of graph representation entails the following steps:

i) First, we obtain feature or corner points which are the nodes of the graph. Here we use the Harris corner detection method discussed in Chapter 1.

ii) Construct Delaunay triangulation on the extracted corner points using Matlab function ***DelaunayTri(x,y)*** which is based on the quickhull (Qhull) algorithm for convex hulls [12].

We note that graphs (known as Delaunay graph) resulting from the Delaunay approach have reduced size compared to graphs resulting from the pixel-based method.
Using the above procedure, we present illustration of Delaunay graph representation of 8 selected objects from the COIL-100 database [85]. The database consists of color images (on dark background) of 100 objects. For each object, different images were taken at every 5 degree turn up to 360, making a total of 72 images per object.
In our experiments, we consider 8 objects and their corresponding images at different views. In order to extract the graph corresponding to each image, we implement the algorithm explained in the previous subsection using my own code in Matlab (available on request) onto the selected objects. The extracted Delaunay graphs are then superimposed onto the objects as shown in Figure 4.11.



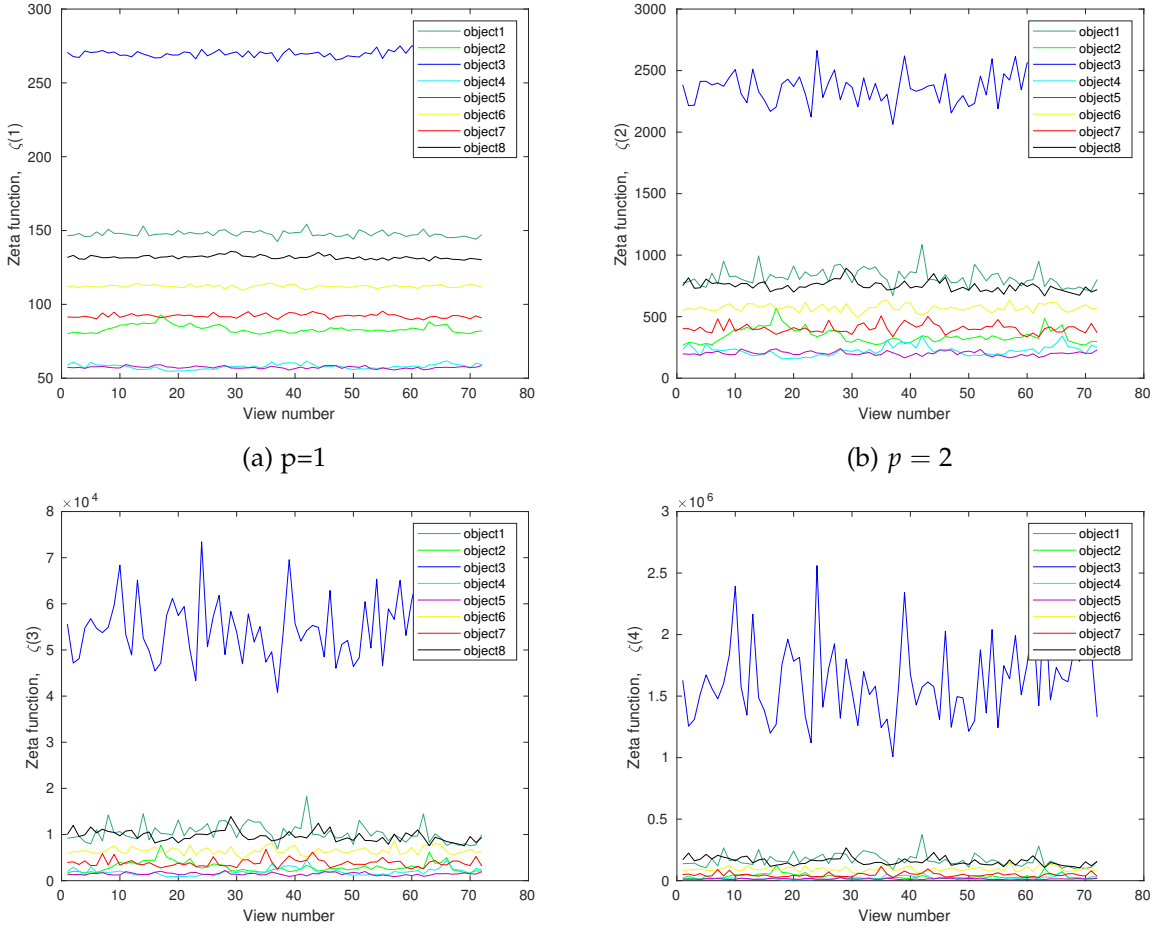
(a) object 1 (b) object 2 (c) object 3 (d) object 4

(e) object 5 (f) object 6 (g) object 7 (h) object 8

Figure 4.11: Illustration of 8 selected objects from the COIL-100 database with their Delaunay graphs superimposed.

4.10.2 The zeta function experiments

Here, we explore the use of the zeta function (equation 4.7.2) as a means of characterising the structure of a graph. For the 8 selected objects from the COIL database, we extract graphs for different images of each object using the method explained in the previous subsection. First, we consider the zeta function associated with the non-zero eigenvalues of the normalised Laplacian matrix, that is to say, we use equation 4.7.2 where μ_i denotes the eigenvalues of the standard normalised Laplacian matrix, that is to say, with no long-range interactions considered. We perform simulations for different values of zeta argument, p, for each of the graphs extracted for different views of each of the 8 selected objects.



(a) p=1

(b) $p = 2$

(c) $p = 3$

(d) $p = 4$

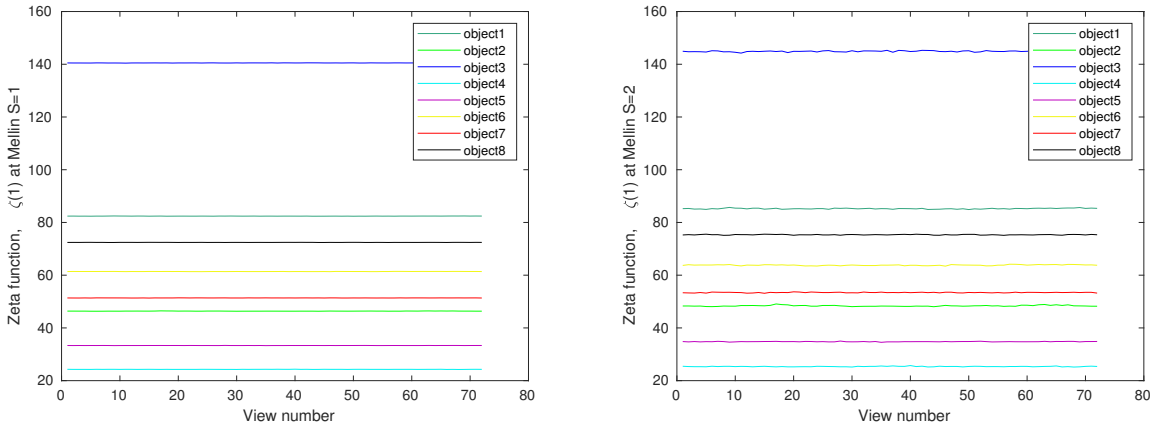
Figure 4.12: Plots of the zeta function $\zeta(p)$ associated with the normalised Laplacian eigenvalues against view number. (from left to right, and top to bottom, $p = 1, 2, 3$ and 4 respectively).

In Figure 4.12, we observe that 8 different curves corresponding to the 8 selected objects are distinct in all the four plots. We, however, see that the curve corresponding to object 3 is much separate (extreme top) from the other 7 objects. This is because, for each view of object 3, the extracted Delaunay graph consists of about 150 vertices which is very large compared

to vertices of the graphs of the other objects which is in the range $40 - 90$. Thus, a large number of non-zero eigenvalues corresponding to object 3 and consequently a large value of the corresponding zeta function.
We then turn our attention to the individual plots, we observe minimal fluctuations in the value of the zeta function of each object at small values of the zeta exponent, p. However, as the value of p increases, we observe that the fluctuations become larger. From the zeta function formula given by equation 4.7.2, it follows that the value of the zeta function is dependent on the p-th inverse of the eigenvalues of the corresponding matrix and thus for large values of p for instance $p = 4$, small eigenvalues become more dominant in determining the value of the zeta function thus responsible for larger fluctuations.

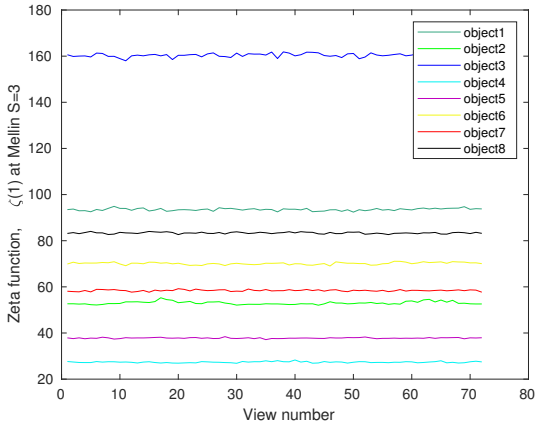
Moreover, we perform similar experiments for the zeta function associated with the eigenvalues of the normalised generalised Laplacian matrix of the graphs corresponding to different views of the 8 selected objects. First, we consider a case for zeta function argument $p = 1$ for different values s and λ and results are shown in Figure 4.13 and Figure 4.15 respectively. From the results, we observe that the curves for different objects are well separated which indicates that the zeta function associated with the generalised Laplacian eigenvalues can also be used to characterise graphs. Furthermore, we observe that as the values of the Mellin (or Laplace) exponents s (or λ) increase, fluctuations in the value of zeta function become large. This can be explained based on equation 3.5.10 where by as the exponents s (or λ) increase, the associated eigenvalues of the generalised Laplacian matrix decrease as they approach the spectrum of the standard Laplacian matrix and thus exponentiating these eigenvalues gives huge magnitudes of the zeta function thus the large fluctuations. For small values of s (or λ), the corresponding spectrum has large eigenvalues which approach the spectrum of the complete graph as s (or λ) tend to zero. When we turn our attention to the difference in the magnitude of fluctuations in the zeta function associated with the Mellin transform-based and Laplace transform-based generalised Laplacian matrices, we observe that fluctuations are relatively smaller in the former than in the latter case for the same value of zeta argument, p, and Mellin or Laplace exponents. For example, Figures 4.13b and 4.15b are plots of the zeta function at exponent, $p = 1$ against view number for $s = 2$ and $\lambda = 2$ respectively. We can see from these figures that fluctuations are larger in the latter case than the former. This is attributed to the fact that the eigenvalues associated with the Mellin transform-based generalised Laplacian matrix are larger in value compared to eigenvalues associated with the Laplace-based generalised Laplacian matrix following from equation 3.5.5 and equation 3.5.4 used to compute the generalised Laplacian matrix in the two cases respectively. In addition, as the values of s and λ decrease, the values of the corresponding eigenvalues become larger which on exponentiating results into smaller values of zeta function compared to the previous large value of s or λ and in addition the dominance of the small eigenvalues is low and this explains the observed trend of increase in the magnitude of fluctuations as s (or λ) increases. On increasing the value of the zeta exponent from $p = 1$ to $p = 2$, we observe that fluctuation become large for the same values s (or λ) in both cases. This behaviour is due to the increased

dominance of the small eigenvalues to the overall value of the zeta function as the value of p increases.

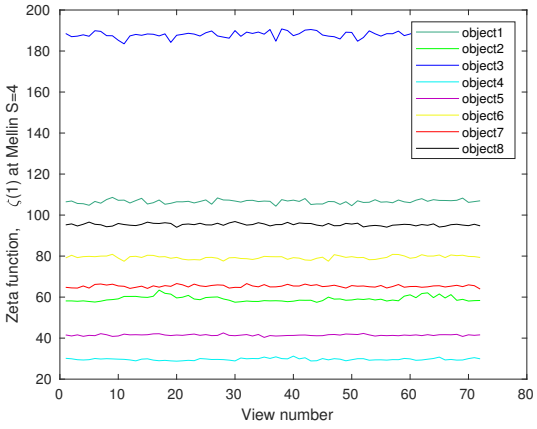


(a) $s = 1$

(b) $s = 2$

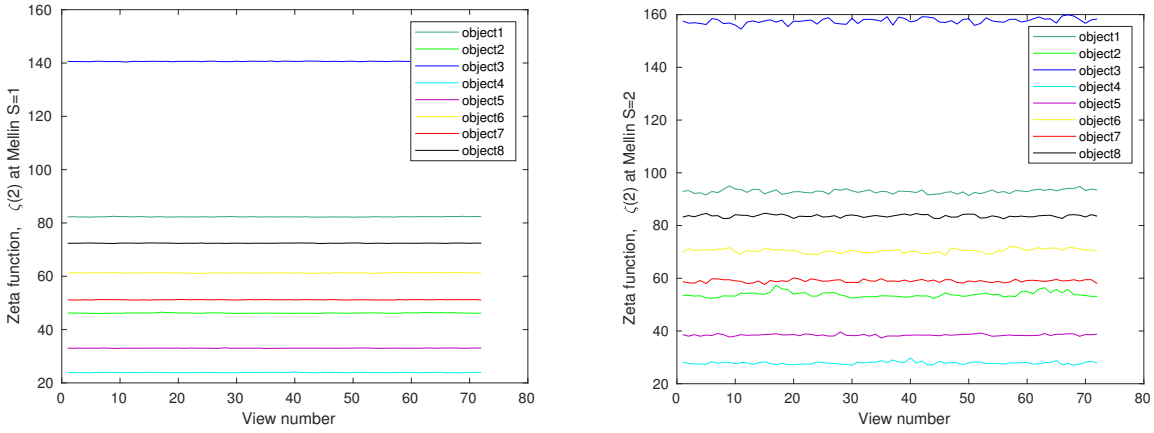


(c) $s = 3$



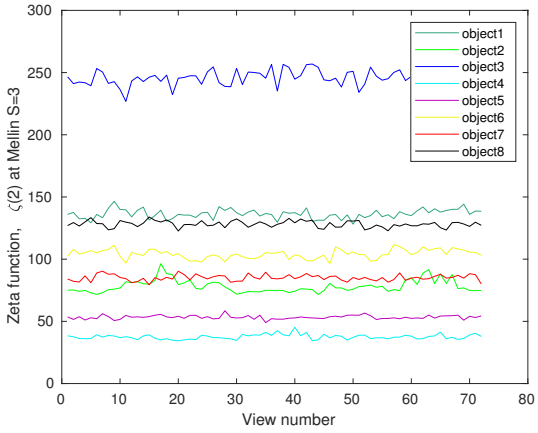
(d) $s = 4$

Figure 4.13: The zeta function $\zeta(p)$ at $p = 1$, associated with the Mellin transform-based generalised normalised Laplacian eigenvalues against view number.

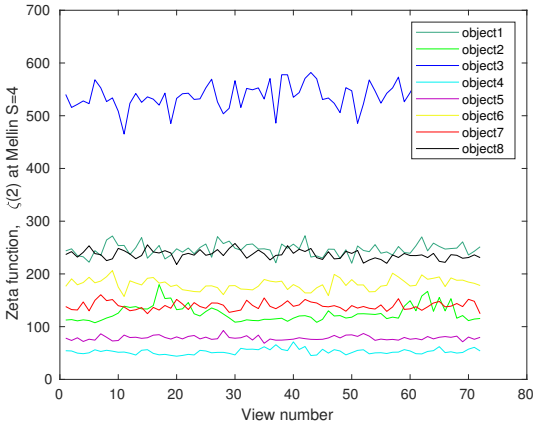


(a) $s = 1$

(b) $s = 2$

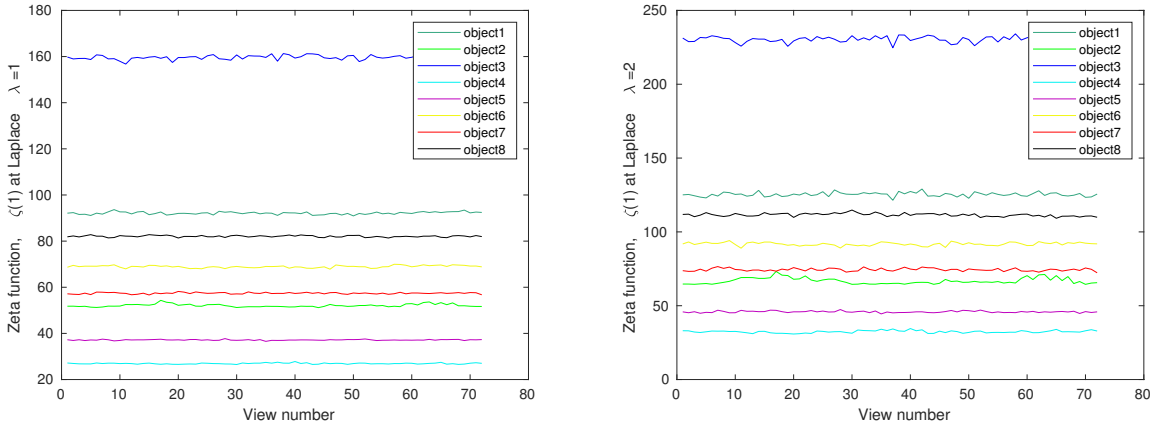


(c) $s = 3$



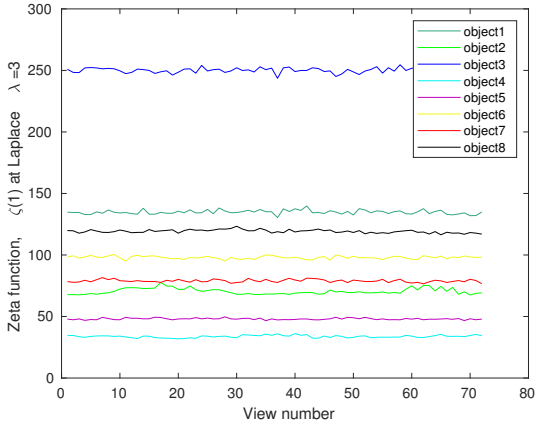
(d) $s = 4$

Figure 4.14: Plots of the zeta function $\zeta(p)$ at $p = 2$, associated with the Mellin transform-based generalised normalised Laplacian eigenvalues against view number.

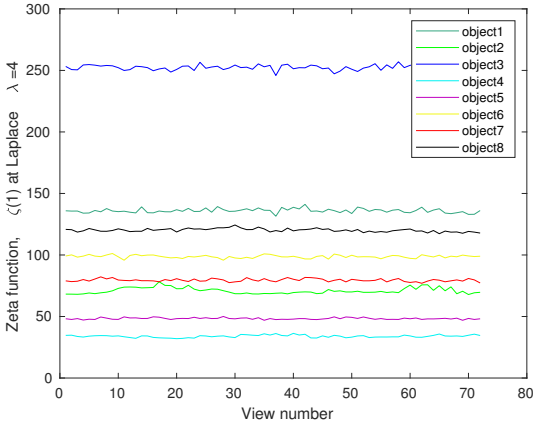


(a) $\lambda = 1$

(b) $\lambda = 2$

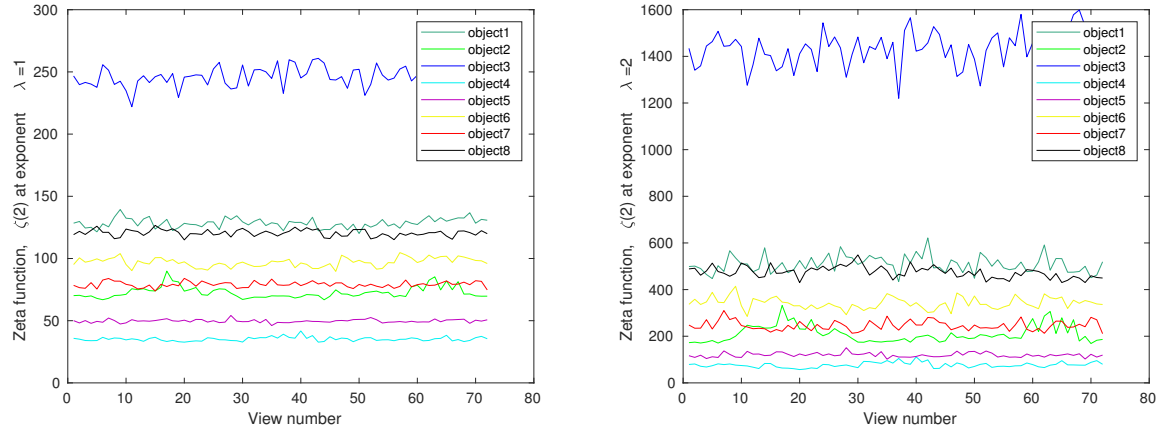


(c) $\lambda = 3$



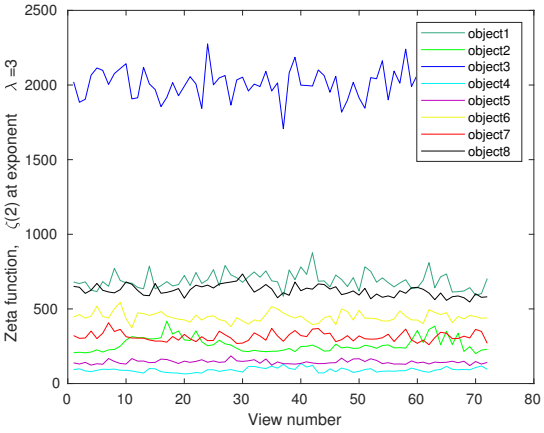
(d) $\lambda = 4$

Figure 4.15: Plots of the zeta function $\zeta(p)$ at $p = 1$ associated with the Laplace transform-based generalised normalised Laplacian eigenvalues with view number.

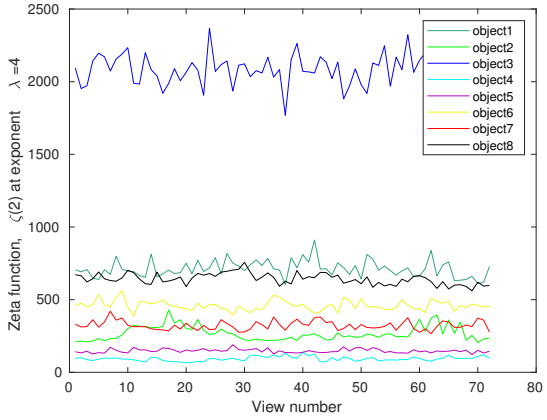


(a) $\lambda = 1$

(b) $\lambda = 2$



(c) $\lambda = 3$

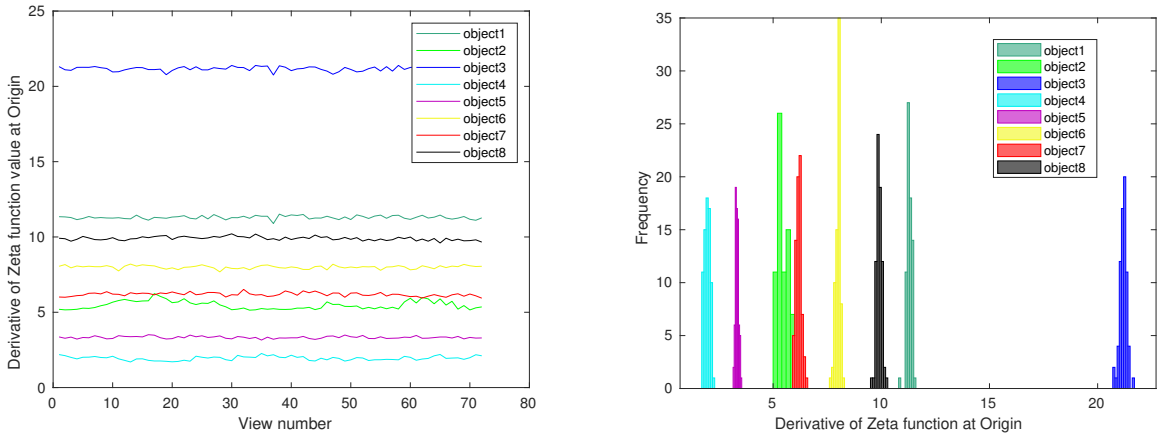


(d) $\lambda = 4$

Figure 4.16: Plots of the zeta function $\zeta(p)$ at $p = 2$ associated with the Laplace transform-based generalised normalised Laplacian eigenvalues with view number.

4.10.3 Experiments of the derivative of the zeta function at origin

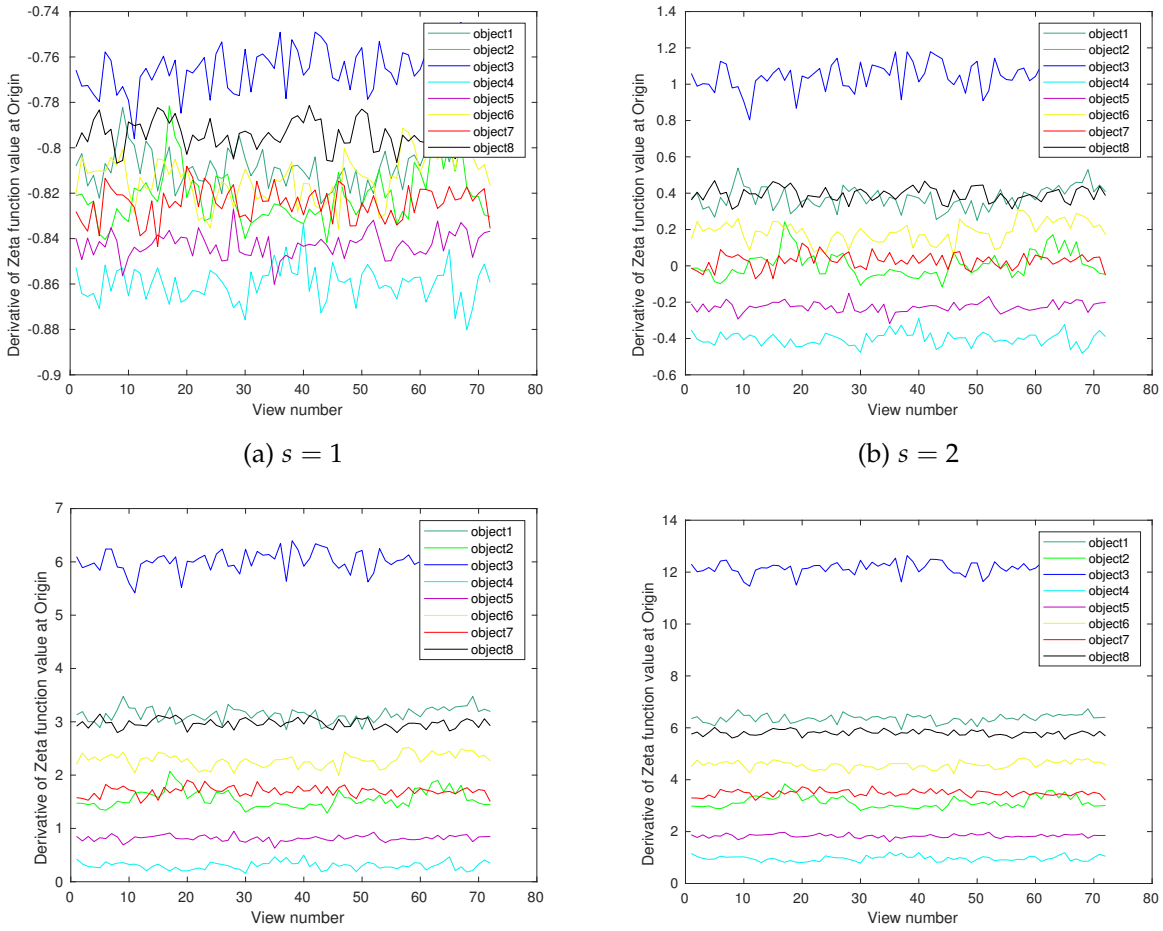
Here, we consider the use of the derivative of the zeta function at origin as a means of graph characterisation. We proceed with similar experiments as we performed with the zeta function and the results are shown in Figure 4.17.



(a) (b)

Figure 4.17: The derivative of the zeta function (associated with the standard Laplacian matrix) at the origin as a function of view number (a) and a histogram of the derivative of the zeta function at the origin for 8 selected objects of the COIL database (b).

We observe from Figure 4.17a that the curves for each of the 8 objects are well separated and the fluctuations are low. More importantly, the curves in this plot are better separated compared to any of the plots of the zeta function in Figure 4.12 which implies that derivative of the zeta function at the origin performs better at differentiating objects than any of the zeta functions for different arguments. Furthermore, in Figure 4.17b, the histogram of $\zeta'(0)$ shows a clear separation of different objects.



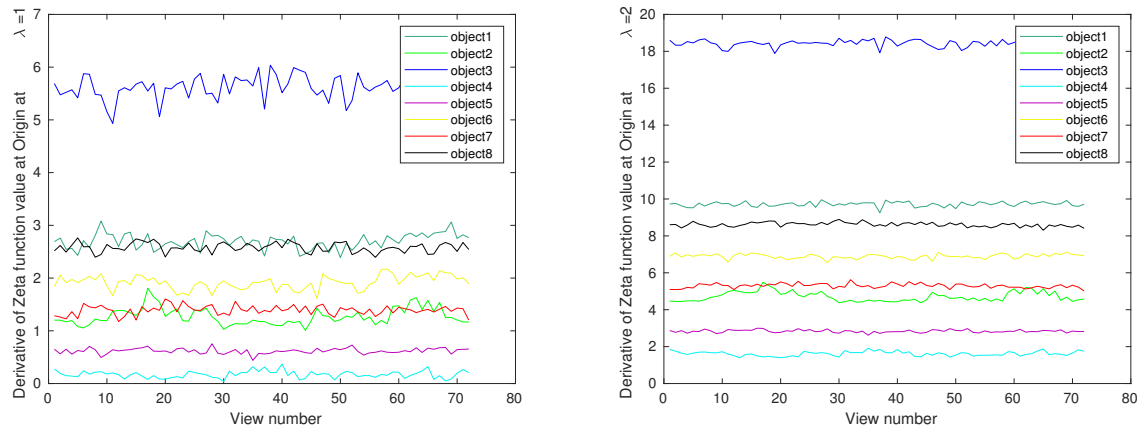
(a) $s = 1$

(b) $s = 2$

(c) $s = 3$

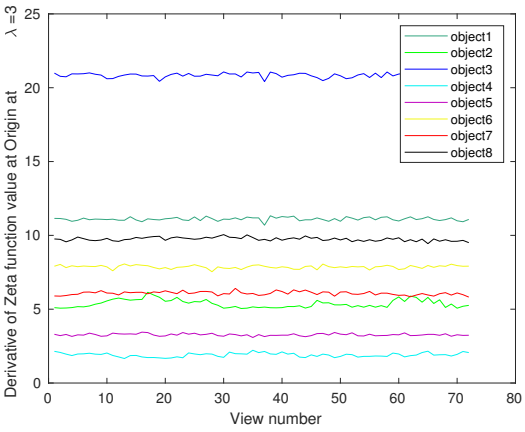
(d) $s = 4$

Figure 4.18: Plot of derivative of the zeta function of the Mellin-based generalised Laplacian matrix at the origin against view number (from left to right, and top to bottom, $s = 1$, 2, 3, and 4 respectively).

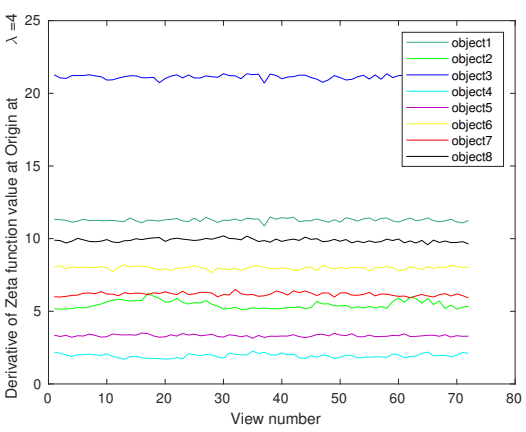


(a) $\lambda = 1$

(b) $\lambda = 2$



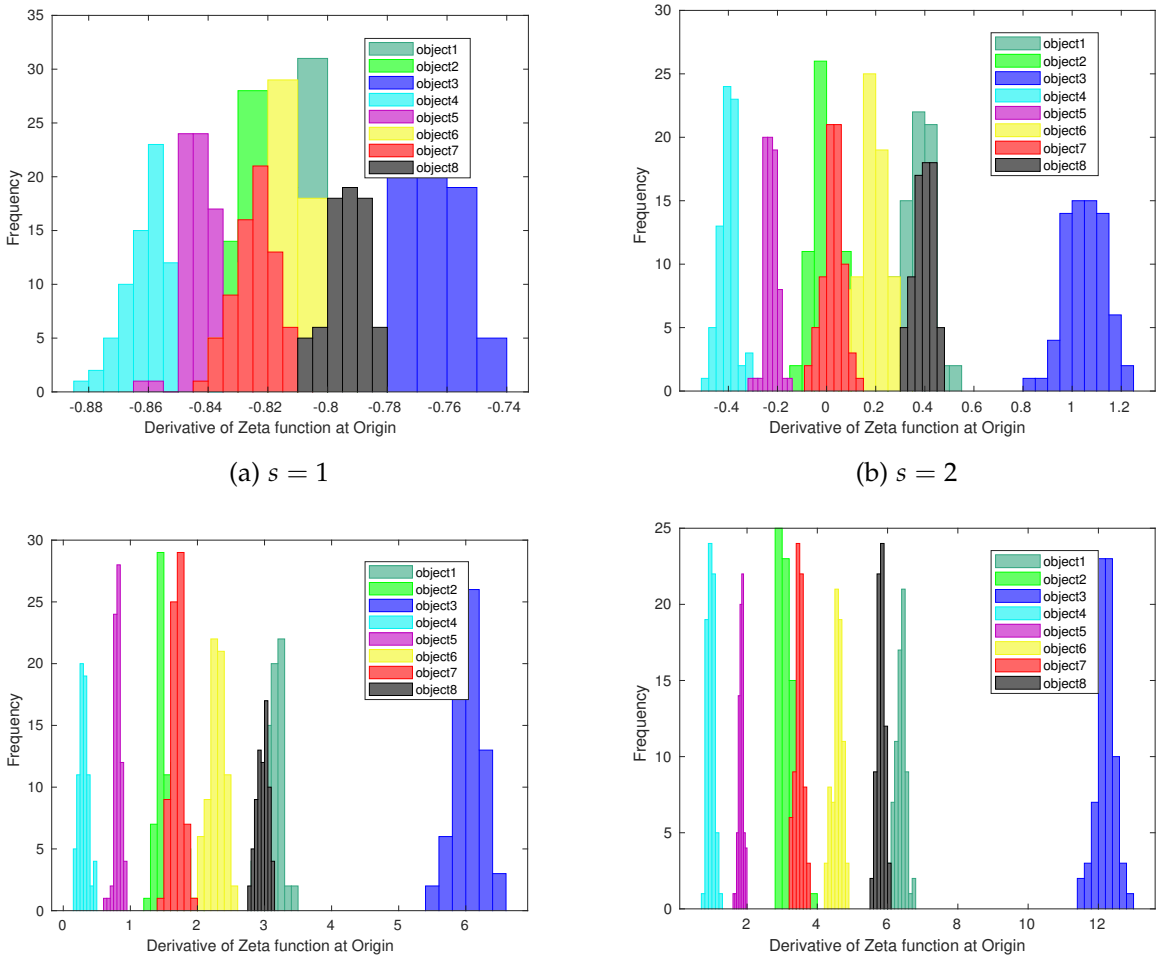
(c) $\lambda = 3$



(d) $\lambda = 4$

Figure 4.19: Plot of derivative of the zeta function of the Laplace-based generalised Laplacian matrix at the origin against view number (from left to right, and top to bottom, $\lambda = 1$, 2, 3, and 4 respectively).

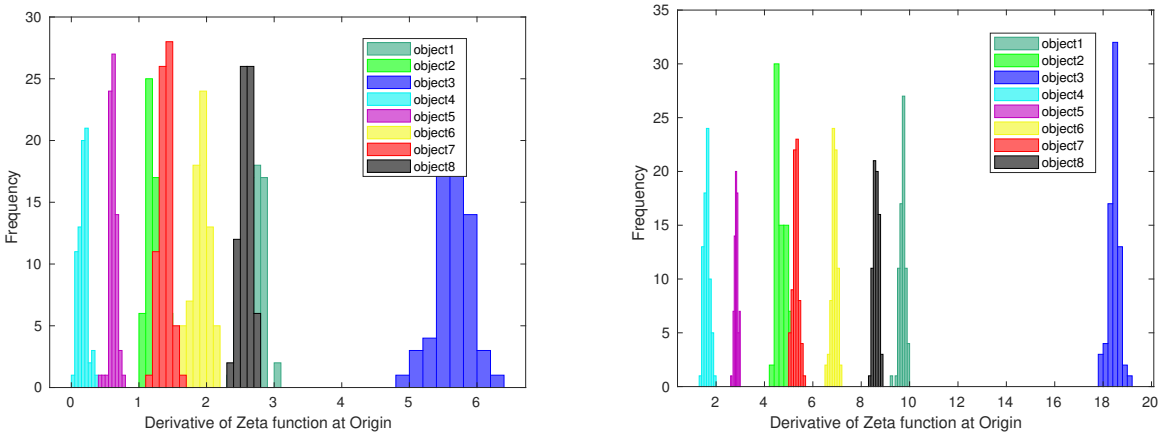
From Figures 4.18 and 4.19, we can see that for small values of exponent s and λ, fluctuations are large and objects are not well separated. As the values of s and λ increase, we observe a decrease in fluctuations and objects become well separated. This is due to the fact that as the values of exponents s or λ increase, the strength of long-range interactions decreases and thus smaller fluctuations and better separation of objects. This behaviour is even more evident in the histograms for different values of s and λ as shown in Figure 4.20 and Figure 4.21 respectively.



(a) $s = 1$ (b) $s = 2$

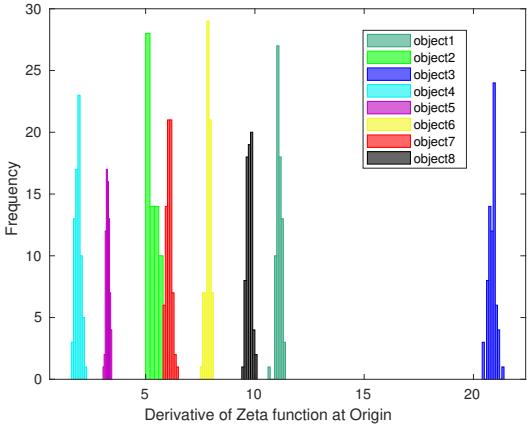
(c) $s = 3$ (d) $s = 4$

Figure 4.20: Histogram of derivative of the zeta function of the Mellin-based generalised Laplacian matrix at the origin against view number (from left to right, and top to bottom, $s = 1$, 2, 3, and 4 respectively).

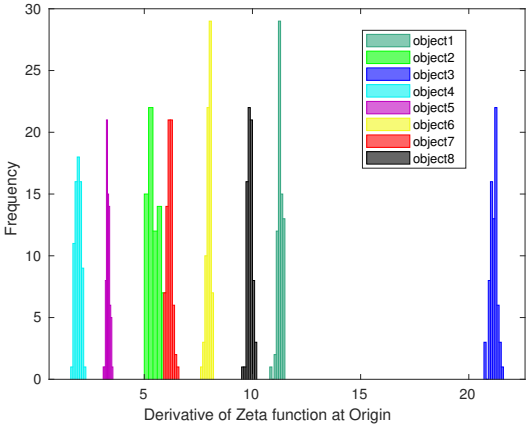


(a) $\lambda = 1$

(b) $\lambda = 2$



(c) $\lambda = 3$



(d) $\lambda = 4$

Figure 4.21: Histogram of derivative of the zeta function of the Laplace-based generalised Laplacian matrix at the origin against view number (from left to right, and top to bottom, $\lambda = 1$, 2, 3, and 4 respectively).

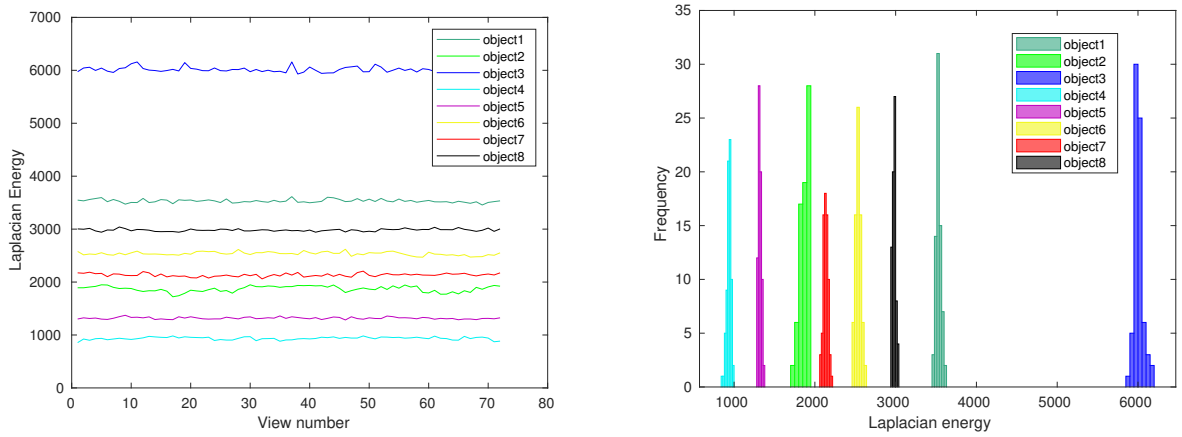
From Figures 4.20 and 4.21, we can see that for small values of exponent s and λ, the corresponding histograms have overlapped bars which indicates that objects are not well clustered. As the values of s and λ increase, we observe that the resulting histograms have well separated bars. We attribute this behaviour to the fact that as the values of exponents s or λ increase, the strength of long-range interactions decreases and thus better separation of objects in the histogram.

4.10.4 Laplacian Energy

The Laplacian energy of a graph is the sum of squares of eigenvalues of its Laplace matrix. It is given by

$$E_L = \sum_{\lambda_i \neq 0} \lambda_i^2 \tag{4.10.1}$$

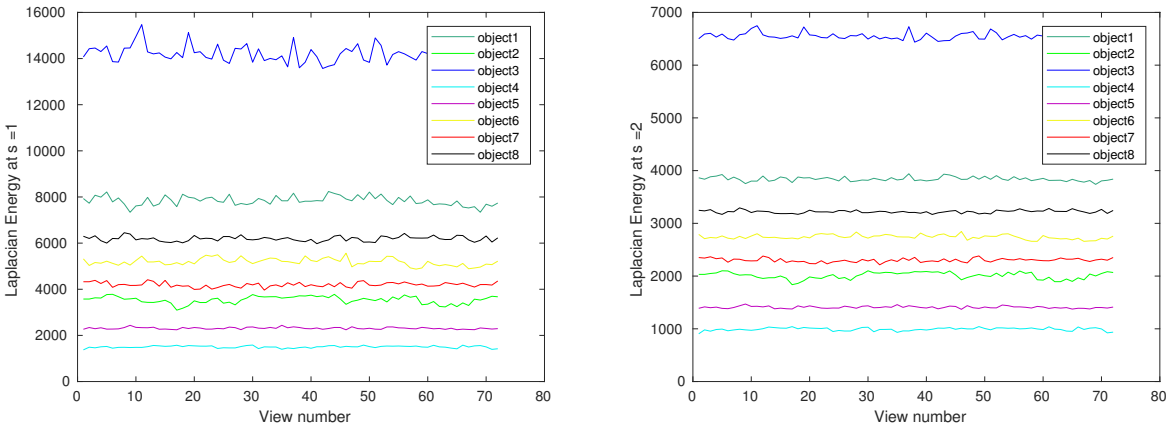
Like the derivative of the zeta function, the Laplacian energy is an invariant to node label permutations and is parameter independent. These properties thus pose a possibility of using the Laplacian energy for purposes of graph characterisation. The results of experiments performed on the eight selected objects of the COIL database using the Laplacian energy of the extracted graphs are shown in Figure 4.22.



(a) (b)

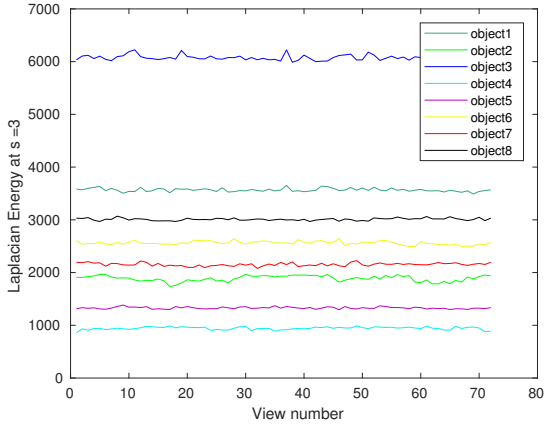
Figure 4.22: Plot of Laplacian energy as a function of view number in (a) and the histogram of the Laplacian energy for 8 selected objects of the COIL database in (b).

From Figure 4.22, we observe that the Laplacian energy performs well in distinguishing different graphs extracted from the objects compared to the derivative of the zeta function at the origin which is also a unitary feature. We further perform similar experiments using the generalised Laplacian energy based on both the Mellin transform and Laplace transforms of the k-path Laplacian matrices as shown in Figures 4.23 and 4.24. The results from the experiments show that Laplacian energy computed from the generalised Laplacian matrix performs well in distinguishing different graphs.

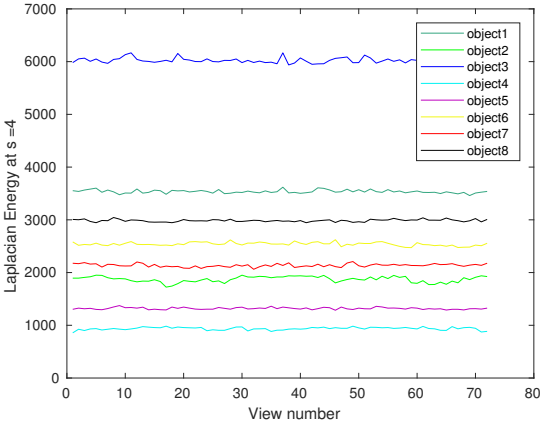


(a) $s = 1$

(b) $s = 2$

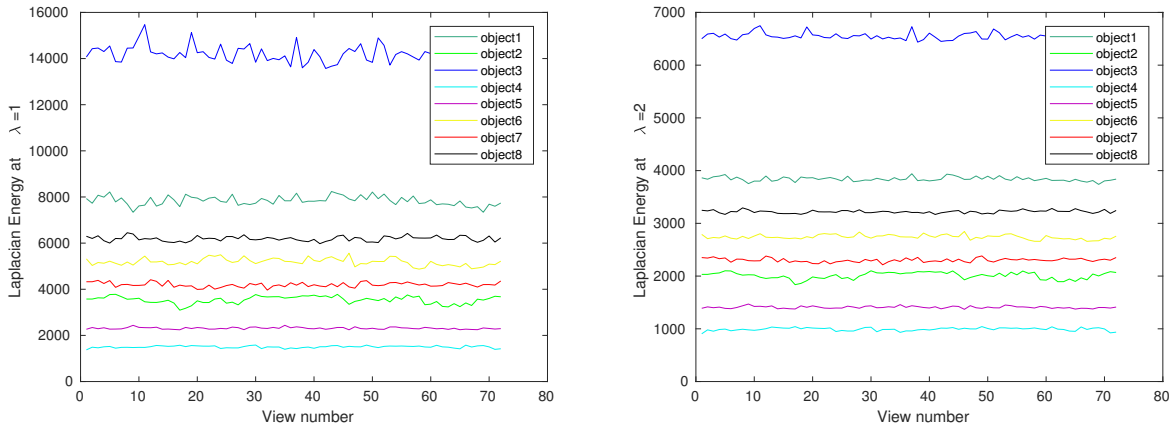


(c) $s = 3$



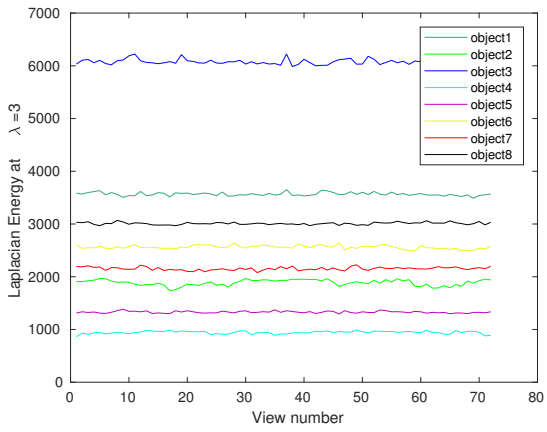
(d) $s = 4$

Figure 4.23: Plot of the Laplacian energy associated with the Mellin-based generalised Laplacian matrix against view number (from left to right, and top to bottom, $s = 1$, 2, 3, and 4 respectively).

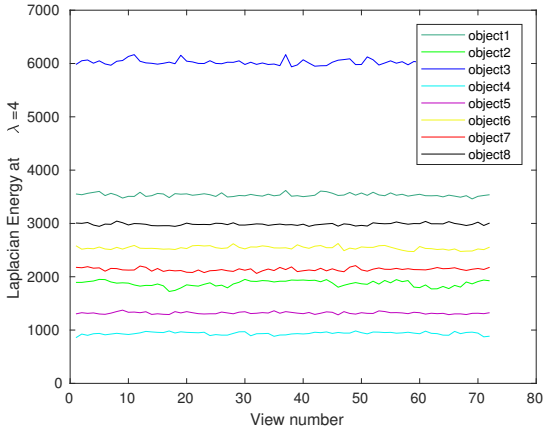


(a) $\lambda = 1$

(b) $\lambda = 2$



(c) $\lambda = 3$



(d) $\lambda = 4$

Figure 4.24: Plot of Laplacian energy associated with the Laplace transform-based generalised Laplacian matrix against view number (from left to right, and top to bottom, $\lambda = 1$, 2, 3, and 4 respectively).

4.11 Graph-based Image clustering

Image clustering is a very important application in many image analysis tasks such as object recognition, object tracking and many others. The task of image clustering involves categorising images based on similarities or differences in various features among them. Such features are based on intensity, colour, shape, texture, among others. In this thesis, we explore clustering of objects using information obtained exclusively from the object itself. In other words, no other information attached to the object such as tags, location, and any other form of additional information is used in the process of object clustering. Recently, it is common practice to use graphs to capture the structure of images in shape-based object clustering. There are various approaches used to extract graphs representing images, however, in this thesis we focus on the Delaunay graph approach explained in the previous section (section 4.10). With the extracted graphs at hand, graph clustering techniques are then applied to them. The fun-

damental step in the process of graph clustering is ascertaining the similarity between graphs such that graphs that are similar to each other are categorised as a cluster or group. There are various approaches to graph clustering which include but not limited to the following:

One approach involves defining a class prototype that captures the most noticeable features of a class. Individual graphs are then compared to respective class prototypes such that those which match particular prototypes are assigned accordingly, thus forming clusters. One of the major problems of this approach is the difficulty of defining the class prototype (a graph that represents each cluster). Fortunately, authors in [21] have attempted to overcome this problem by developing generic algorithms that generate class prototypes for different clusters. The other problem is dealing with graphs of different numbers of nodes. Another approach is known as pairwise clustering. Given a set of graphs, pairwise clustering involves a comparison between pairs of graphs normally based on the edit distances of the two graphs. The resulting clusters consist of subsets of the graph set that have strong mutual pairwise affinities [77, 105].

One drawback of the two approaches is that correspondence between nodes must be computed beforehand which is a computationally cumbersome task especially for large graphs. To overcome this problem, methods have been developed which use feature vectors. It is noteworthy that pattern vectors can be easily manipulated compared to graphs. However, difficulties arise in vectorisation of graphs which include: First, there is no natural way of mapping nodes or edges of a graph into components of vectors as there exists no standard ordering of nodes or edges of graphs. Second, suppose we find an ordering of nodes or edges, vectors of different lengths can be obtained for graphs with different number of nodes and edges [78, 116]. This thus calls for a method of dealing with pattern-vectors of different lengths. The difficulty associated with vectorising graphs makes it practically impossible to adopt clustering methods which involve pairwise similarity measurements. Luo et al.[78] attempted to overcome this problem by using the spectral representation of graphs in which the structure of a graph is mapped onto a vector of fixed length. The pattern vectors are developed based on the adjacency matrix of a graph where the content of the vectors are features such as leading eigenvalues, eigenmode volume, eigenmode perimeter and Cheeger constant.

Alternatively, the vectors can be composed of binary features which are pairwise attributes of eigenmodes and these include mode association matrix and inter-node distances. Shortly after, Xiao [118] introduced mapping of graphs onto pattern vectors using the spectrum of the Laplacian matrix as well as other invariants of the heat kernel such as the zeta function and coefficients of heat content polynomials. The vectors of invariants can then be embedded into low-dimensional space using techniques such as Principal Component Analysis (PCA) [98], Multi-Dimensional Scaling (MDS) [104], Locality Preserving Projection (LPP) [60], and many others.

In this section, we explore how pattern vectors are constructed from invariants of the generalised heat kernel. We further present results of experiments of graph clustering based on these invariants. Moreover, we discuss the impact of long-range interactions in graph clustering.

4.11.1 Constructing pattern vectors of objects

One very important and essential step in clustering of objects is extraction of image features. For good clusters to be attained, features used must be appropriate, that is to say, they should capture the important information of an image. Moreover, since images are of very high dimensions, the features must allow for reduction in dimension so as to make the features appropriate as input to clustering algorithms. One thing worth noting is that in order to capture complete information of images, methods that involve appropriate combination of features have been developed since no particular method based on a single feature performs well in completely capturing the important information of an image as discussed in the review in [31]. It was observed that particular image descriptors or features perform well at specific tasks or for specific image databases [31].
There are various methods used in constructing pattern vectors most of which are based on the spectrum of the adjacency or Laplacian matrices of corresponding graphs. We discuss some of these methods:

4.11.1.1 Leading eigenvalues

For a graph indexed k, that is, G_k extracted from a particular object, a feature vector that represents G_k can be constructed from ordered eigenvalues of the (normalised) Laplacian matrix of a graph. We recall that the Laplacian eigenvalues are invariant to node label permutations. The feature vector constructed from the spectrum is given as $\boldsymbol{B}_k = (l_1, l_2, \ldots, l_n)^T$ where l_i is the i-th eigenvalue. The eigenvalues follow a non-increasing order: $l_1 \geq l_2 \geq \ldots \geq l_n$. Since in our experiments we are dealing with 72 views of each of the 8 objects, we will use the 6 leading eigenvalues, that is, $n = 6$ to reduce on complexity in computation.

4.11.1.2 The zeta function at different values of argument

Xiao [118] showed that the zeta function associated with the Laplacian eigenvalues can be used in constructing feature vectors. The vector is composed of zeta function values at different arguments. That is, for a graph G_k, the corresponding feature vector
$\boldsymbol{B}_k = (\zeta(1), \zeta(2), \ldots, \zeta(n))^T$. Again for computation convenience, we will take $n = 4$ in the experiments.

4.11.1.3 Symmetric polynomials

Symmetric polynomials are polynomials on a set of variables that are invariant under permutation of variable indices. In other words, on interchange of one or more variables, one

obtains the same value of the polynomial. Due to this invariant property, symmetric polynomials have proven useful in constructing pattern vectors used for graph analysis [116, 118]. Given n variables $x_1, x_2, \ldots, x_n$. A polynomial $S \in k[x_1, x_2, \ldots, x_n]$ is symmetric if

$$S(x_1, x_2, \ldots, x_n) = S(x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(n)}), \tag{4.11.1}$$

for all possible permutations $x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(n)}$ of the variables.

There are various types of symmetric polynomials, however, in this thesis we focus on two of them namely the elementary and power-sum symmetric polynomials.

1. Elementary symmetric polynomials: For a given set of n variables $\{x_1, \ldots, x_n\}$, the set of elementary symmetric polynomials are given by

$$\begin{aligned}
S_1(x_1, \ldots, x_n) &= \sum_{i=1}^{n} x_i \\
S_2(x_1, \ldots, x_n) &= \sum_{i=1}^{n} \sum_{j=i+1}^{n} x_i x_j \\
&\vdots \\
S_k(x_1, \ldots, x_n) &= \sum_{i_1 < i_2 < \ldots < i_k} x_{i_1} x_{i_2} \ldots x_{i_k} \\
&\vdots \\
S_n(x_1, \ldots, x_n) &= \prod_{i=1}^{n} x_i
\end{aligned}$$

2. Power-sum symmetric polynomials:

   With the same set of variables as above, the set of power symmetric polynomials are defined as

$$\begin{aligned}
P_1(x_1, \ldots, x_n) &= \sum_{i=1}^{n} x_i \\
P_2(x_1, \ldots, x_n) &= \sum_{i=1}^{n} x_i^2 \\
&\vdots \\
P_k(x_1, \ldots, x_n) &= \sum_{i=1}^{n} x_i^k \\
&\vdots \\
P_n(x_1, \ldots, x_n) &= \sum_{i=1}^{n} x_i^n
\end{aligned}$$

Using the Newton-Girard formula of symmetric functions, the two polynomials are related to each other by the the formula

$$S_k = \frac{(-1)^{k+1}}{k} \sum_{r=1}^{k} (-1)^{r+k} P_r S_{k-r}, \tag{4.11.2}$$

where S_k is shorthand for $S_k(x_1, \ldots, x_n)$ as P_k is for $P_k(v_1, \ldots, v_k)$. The elementary symmetric polynomials S_k can be computed efficiently using the poser symmetric polynomials using the above relation.

For graph clustering purposes, the set of variables to these polynomials can be the n leading eigenvalues of the normalised Laplacian [118], the zeta function for different non-integer arguments [118], the elements of the spectral matrix [116], and many others. Experiments for graph clustering using pattern vectors obtained from symmetric polynomials demonstrate results of good clustering as discussed in [116].

It is worth noting that when the eigenvalues of the (generalised) Laplacian are used as arguments to the symmetric polynomials, the following relations hold:

1. The lowest order elementary symmetric polynomial is the sum of the eigenvalues, that is, the trace of the (generalised) Laplacian.

2. The highest order elementary symmetric polynomial is the product of non-zero eigenvalues of the (generalised) Laplacian and thus related to the derivative of the zeta function at the origin by
$$S_{|V|}(\lambda_1, \lambda_2, \ldots, \lambda_{|V|}) = e^{(\zeta'(0))^{-1}} \tag{4.11.3}$$

3. The second lowest order power symmetric polynomial is the sum of the squares of the eigenvalues which is equal to the Laplacian energy of corresponding graph.

Using symmetric polynomials, feature vectors can be constructed in a number of ways: One is using the vector of leading eigenvalues of Laplacian as an input to the set of symmetrical polynomials. The result of each polynomial then constitutes a component of the feature vector.
Wilson et al. [116] presented a method of constructing the feature vector by using the elements of the spectral matrix (equation 2.3.9). Here, each column of the spectral matrix forms the input to a set of symmetrical polynomials. This results in a set of n^2 spectral features formed by a combination of n columns of the spectral matrix and n symmetric polynomials. It is worth noting that this method is very powerful since it makes use of the full spectral matrix, that is, both eigenvalues and eigenvectors. For graphs of different sizes of node-sets, authors in [116] tackled this problem as follows: For graphs whose node-set size is less than the maximum among a set of graphs, their spectrum are modified by appending eigenvalues of zeros and their corresponding eigenvectors consisting of zeros such that the size of the spectrum of all

graphs is the same, that is to say, equal to that corresponding to graph(s) with the maximum size of nodes [116].

4.11.1.4 Coefficients of heat content polynomials

Since for each graph, the set of coefficients of the heat content polynomial given by equation 4.9.4 is always unique, these coefficients can thus be used for graph characterisation. We can then construct feature vectors whose components are the coefficients indexed by m. That is to say, $\boldsymbol{B}_k = (q_0, q_1, \ldots, q_n)^T$. To ensure less computation overheads, we take the k leading coefficients.

4.11.2 Principal Component Analysis (PCA) on images

As mentioned earlier, PCA is a widely used statistical tool for dimension reduction. The PCA-based dimension reduction relies on selection of dimensions with the largest variance. Ding and He [34] showed that PCA dimension reduction indirectly performs clustering according to the K-means objective function by proving that the principal components are the continuous solution of the cluster membership indicators in the K-means clustering method. In this subsection, we present the steps followed in performing PCA-based dimension reduction on a given number, m, of objects.

i) We commence by selecting the objects from a database to which PCA is to be applied, say m objects.

ii) Next, we extract graphs from each object using a combination of Harris corner detection and Delaunay triangulation techniques as discussed in subsection 4.10.1. This results into graphs $G_1, G_2, \ldots, G_m$.

iii) We construct the feature vector, $\boldsymbol{B}_k$, for each graph, G_k, as discussed in the previous subsection.

iv) We compute the matrix $\boldsymbol{S} = [\boldsymbol{B}_1|\boldsymbol{B}_2|\ldots|\boldsymbol{B}_m]$. The feature vectors form the columns of $\boldsymbol{S}$.

v) We then compute the matrix, $\hat{\boldsymbol{S}}$, by subtracting the mean of the feature vectors from each of the column of the matrix $\boldsymbol{S}$.

vi) We follow this with computation of the covariance matrix $\boldsymbol{C}$ by taking the matrix product $\boldsymbol{C} = \hat{\boldsymbol{S}}\hat{\boldsymbol{S}}^T$.

vii) We extract the principal components directions by first performing eigendecomposition on the covariance matrix, $\boldsymbol{C}$, that is

$$\boldsymbol{C} = \sum_{i=1}^{m} \lambda_i \boldsymbol{v}_i \boldsymbol{v}_i^T, \tag{4.11.4}$$

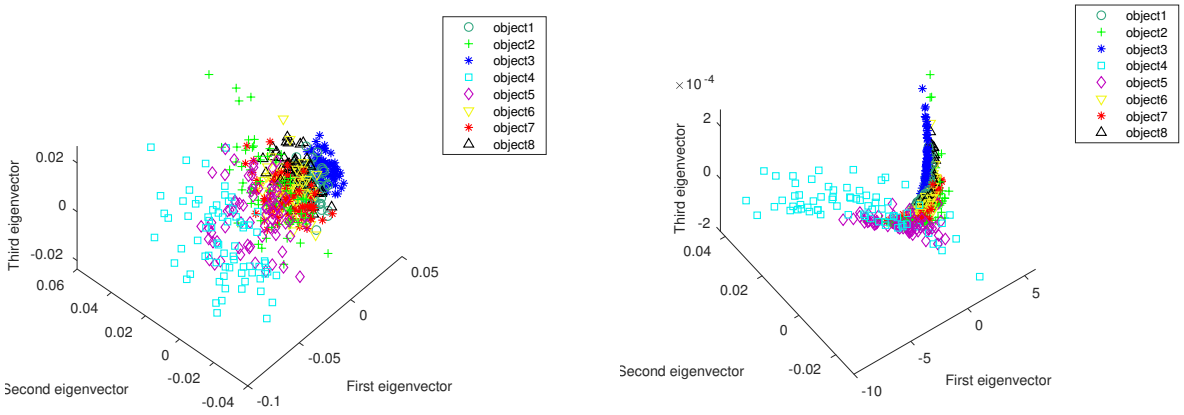
where λ_i are the eigenvalues and $\boldsymbol{v}_i$ are the eigenvectors. This is followed by selection of the first s leading eigenvectors (normally 3 for purposes of visualisation) to represent

the graphs extracted from the objects. By selecting the principal components, we reduce the dimension of the data. The coordinate system of the eigenspace is spanned by the s orthogonal vectors $\boldsymbol{V} = (v_1, v_2, \ldots, v_s)$.

viii) Finally, we project individual graphs (represented by $\boldsymbol{B}_k$ for $1 \leq k \leq m$) onto this eigenspace using $\boldsymbol{B}'_k = \boldsymbol{V}^T \boldsymbol{B}_k$. Therefore, each graph G_k is represented by an s-component vector $\boldsymbol{B}'_k$ in the eigenspace.

4.12 Object clustering experiments

In this subsection, we use the heat kernel invariants to construct feature vectors used for graph clustering on the selected objects of the COIL database described in section 4.10. To begin with, we will use the 6 leading eigenvectors of the normalised Laplacian of each graph to create feature vector. We will then consider these eigenvalues as inputs to the set of elementary symmetric polynomials forming feature vectors. Next, we will show results for using the zeta function associated with the eigenvalues of both the standard Laplacian matrix and the generalised Laplacian matrix. The latter case involves results for different exponents of the Laplace and Mellin transforms of the Laplacian matrix. We will also show results based on the coefficients of the heat content polynomials defined in equation 4.9.4. For visualisation purposes, we perform dimensionality reduction to only 3 dimensions.



(a)

Figure 4.25: Clustering of the 8 objects of Figure 4.11 using PCA with feature vectors composed of the six leading eigenvalues of the graph Laplacian matrix (left) and elementary symmetrical polynomial with the six leading eigenvalues of the normalised Laplacian matrix as variables (right).

In the left-hand panel in Figure 4.25, we observe that PCA performed on the vectors of six leading eigenvalues of the Laplacian matrix results in clusters that overlap which implies that different views of the eight objects are not well classified in this case. On the other hand, the right-hand panel in Figure 4.25, shows that with the six leading eigenvalues as variables to

the set of elementary symmetric polynomials, we observe that the different views of the eight objects do overlap but not as much as in the former case. This implies that using symmetrical polynomials results in relatively better clusters than when the leading eigenvectors are used as feature vectors.

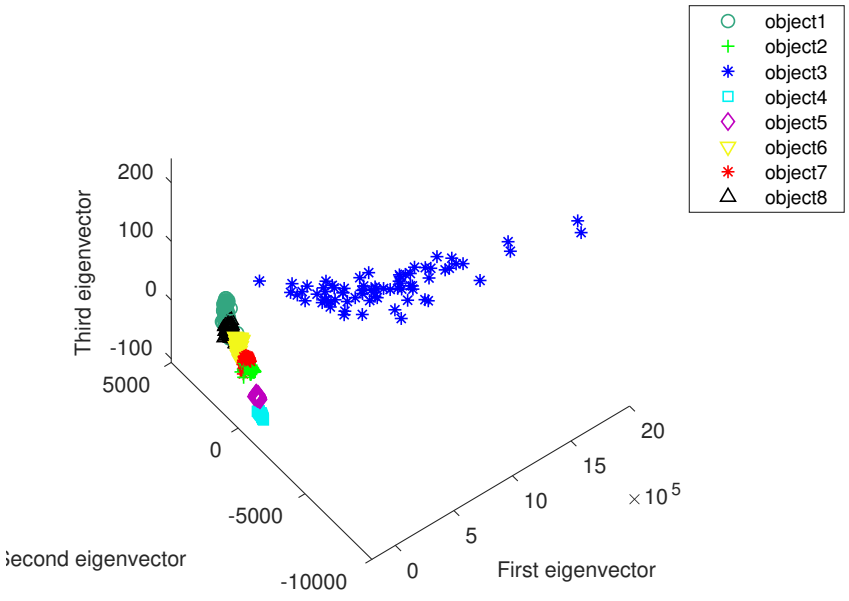


Figure 4.26: Clustering of the 8 selected objects of Figure 4.11 using zeta function for four different integer arguments, that is $p = 1, 2, 3$ and 4.

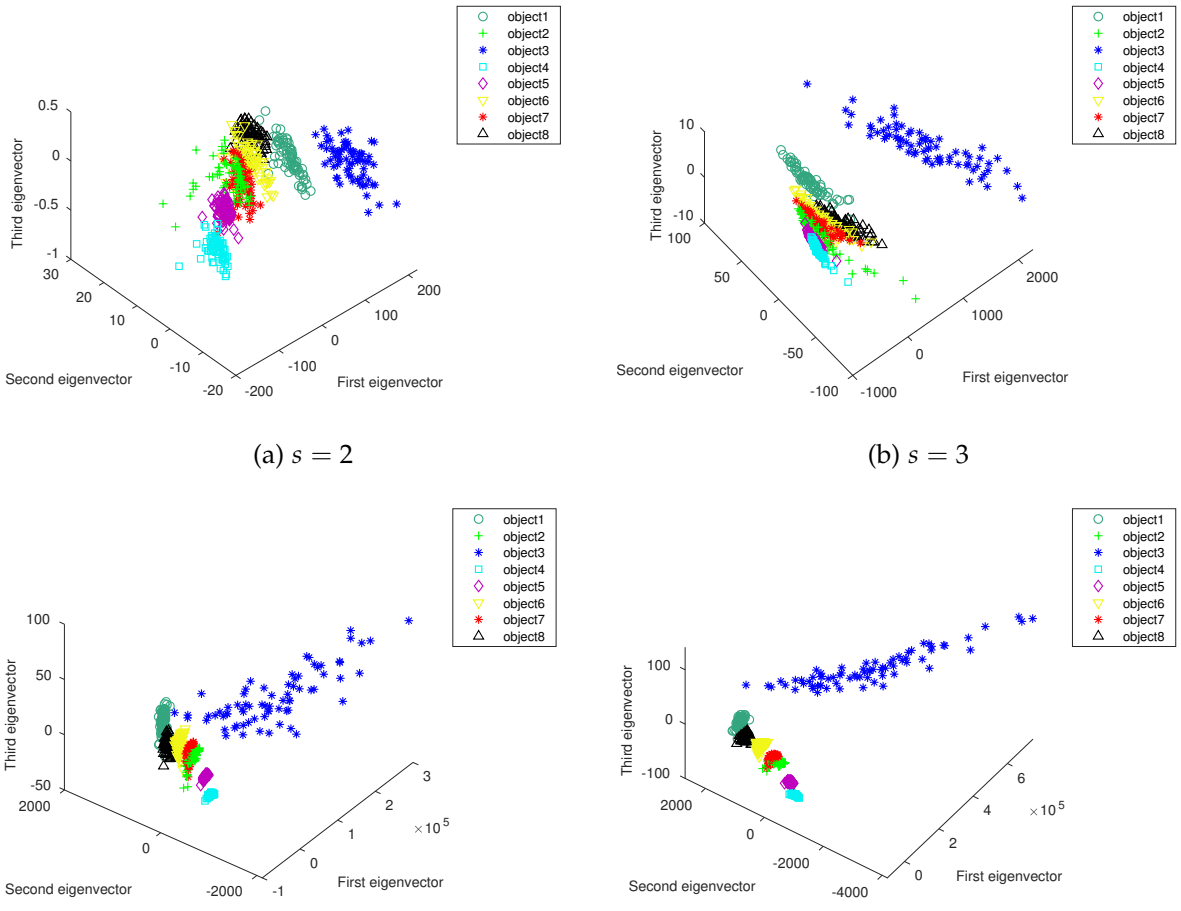
In Figure 4.26, the objects are well separated and the clustering is better when the feature vectors consist of the zeta function for different arguments compared to the one obtained using the leading eigenvalues of the Laplacian shown in Figure 4.25.

We repeat similar experiments with feature vectors constructed from the zeta function associated with the eigenvalues of the generalised Laplacian matrix for the eight objects. In this case, long-range interactions are accounted for by the Mellin and Laplace transforms of the k-path Laplacian matrices. For different exponents, s, of the Mellin and, λ, of the Laplace transform, the results are shown in Figure 4.27 and Figure 4.28 respectively.

First, we note that an increase in the Mellin (or Laplace) exponent, s (or λ), implies a decrease in strength of long-range influence, for instance when $s(\text{or } \lambda) = 2$, the long-range influence is much greater than when $s(\text{or } \lambda) = 6$. On performing clustering using feature vectors constructed from the zeta function of the generalised Laplacian matrix (see Figure 4.27 and Figure 4.28), we observe that for small values of s (or λ), we obtain well-separated clusters and the size of different clusters are big and as the values of s (or λ) increase, the individual clusters become smaller in size. In addition, for object 3, we observe small variation in

its cluster at small values of s and as s increases, so does the variations. For the Laplace transform-based case, however, we observe very small variations in the cluster corresponding to object 3 as λ increases. Generally, as s (or λ) increases, the resulting clusters tend to be similar to the clusters obtained by performing PCA on the zeta function associated with the standard Laplacian matrix (that is, at s (or λ) $= \infty$ which is the case where the influence of long-range interactions is negligible).

This can be explained based on the fact that increase in the value of exponents implies a decrease in strength of long-range interactions and thus clustering obtained tends to the clustering with no long-range interactions.



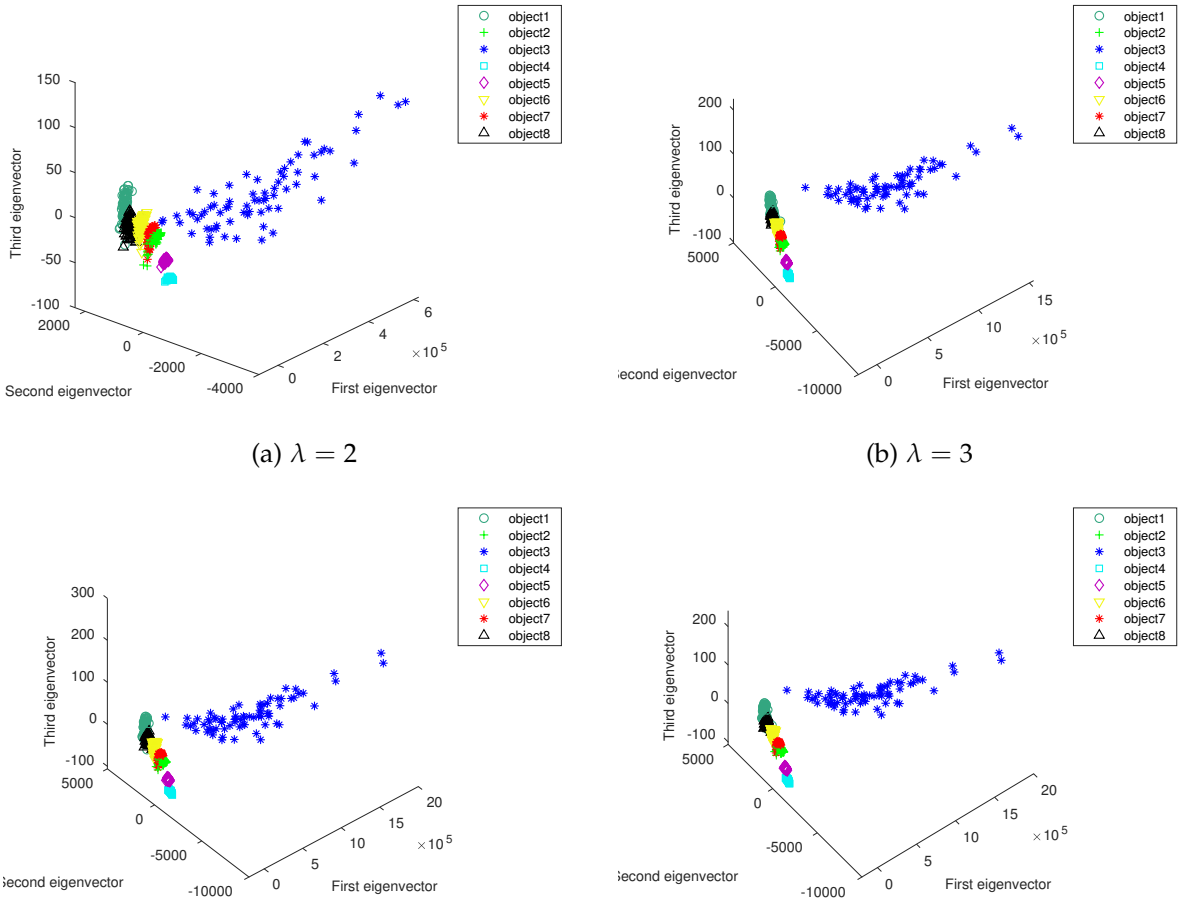
(a) $s = 2$

(b) $s = 3$

(c) $s = 5$

(d) $s = 6$

Figure 4.27: Illustration of PCA based clustering for 8 selected objects of the COIL-100 database. The feature vector consist of the zeta function on 4 arguments ($p = 1, 2, 3, 4$) of the normalised Laplacian matrix of the respective graphs. From left to right and top to bottom, we start with the standard Laplacian followed by generalised Laplacian based on Mellin transform for $s = 2$, 3, 5, and 6.



(a) $\lambda = 2$

(b) $\lambda = 3$

(c) $\lambda = 4$

(d) $\lambda = 6$

Figure 4.28: Illustration of PCA-based clustering for 8 selected objects of the COIL-100 database. The feature vector consist of the zeta function on 4 arguments ($p = 1, 2, 3, 4$) of the normalised Laplacian matrix of the respective graphs. From left to right and top to bottom, we commence with the standard Laplacian followed by generalised Laplacian based on Laplace transform for $\lambda = 2, 3, 4$, and 6.

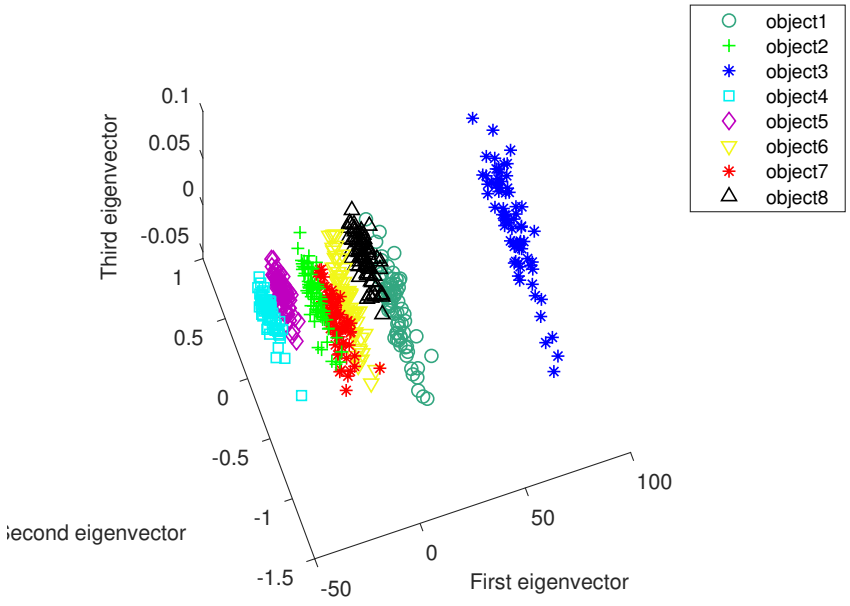
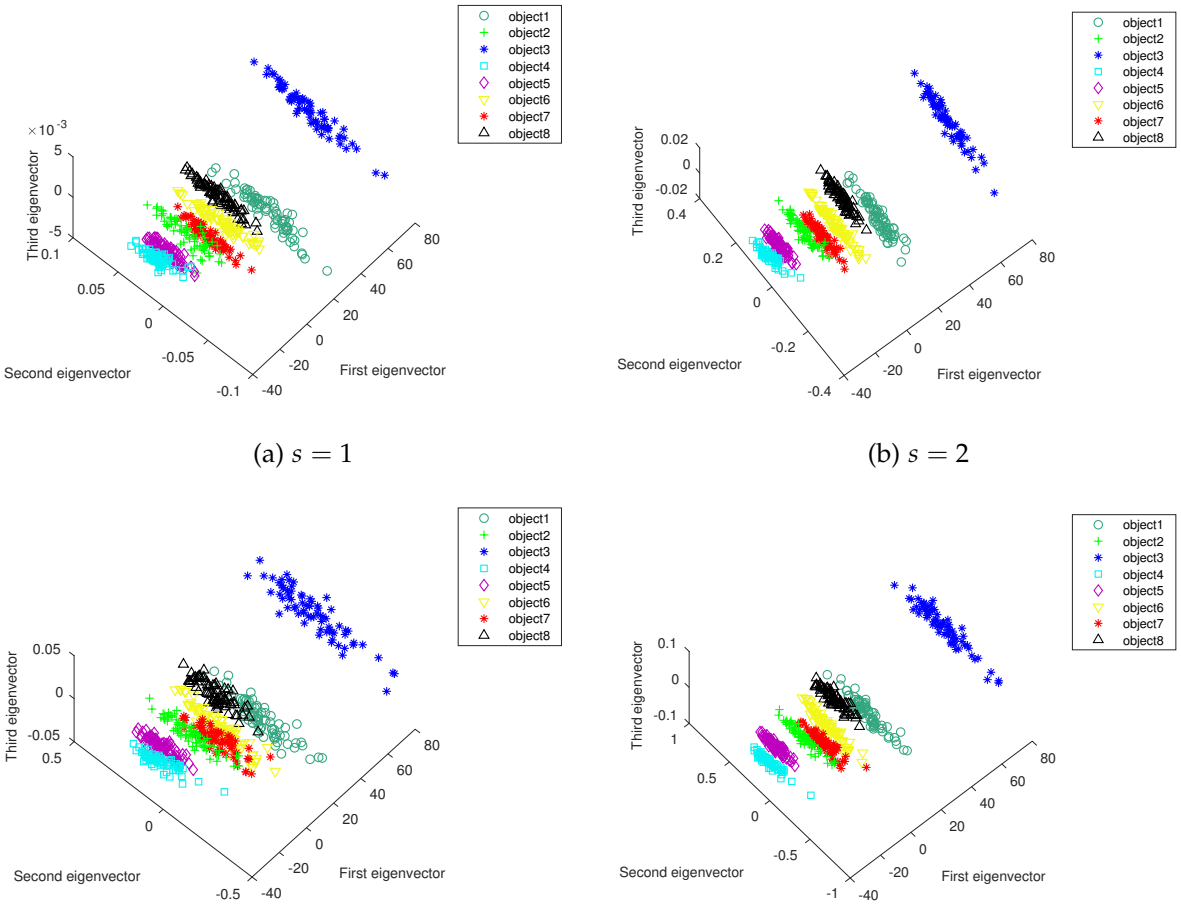


Figure 4.29: Object clustering with feature vectors consisting of four leading coefficients of the heat content polynomials q_0, q_1, q_2, q_3.

As mentioned earlier on, we can perform PCA on the feature vectors constructed from the k leading coefficients of the heat content polynomials. In our experiments, we take $k = 4$ and the results of clustering are illustrated in Figure 4.29. In comparison with Figure 4.25, we observe that analysis of coefficients results in better separation of objects than analysis using the leading eigenvalues of the Laplacian matrix and the zeta function. The better clustering observed using the polynomial coefficients is attributed to the fact that these coefficients are computed using both the eigenvalues and eigenvectors of the Laplacian which completely capture the structure of a graph. On the other hand, eigenvalues of the Laplacian matrix partially capture the structure of a graph and thus invariants constructed using the eigenvalues do not perform well in graph clustering compared to the former case.

We then turn our attention to the clustering using feature vectors constructed from the heat content polynomial coefficients associated with the generalised Laplacian matrix formed by either the Mellin or Laplace transforms of the k-path Laplacian matrices. The results of the simulations are shown in Figure 4.30 and Figure 4.31 for the Mellin transform and Laplace transform-based cases respectively. From the figures, we observe that as the exponents s (or λ) increase, there is a slight difference in the clusters obtained. For small values of s (or λ), clusters are slightly better comparable to higher values of the exponents. The observed trend in the resulting clusters is due to the slight variation in the eigenvectors of the generalised Laplacian matrix as the values of s (or λ) varies and since both eigenvalues and eigenvectors constitute the computation of the heat content polynomial coefficients, the slight variation in the latter is possibly responsible for the observed slight variation in clusters obtained.



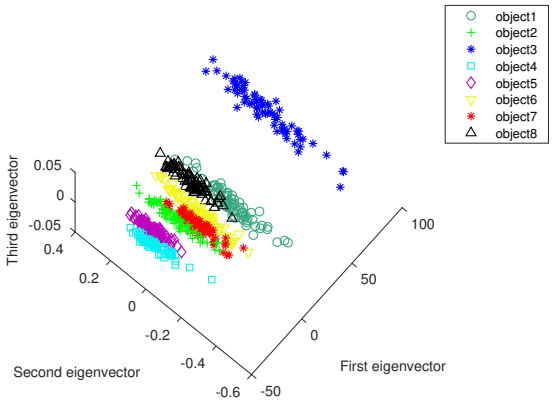
(a) $s = 1$

(b) $s = 2$

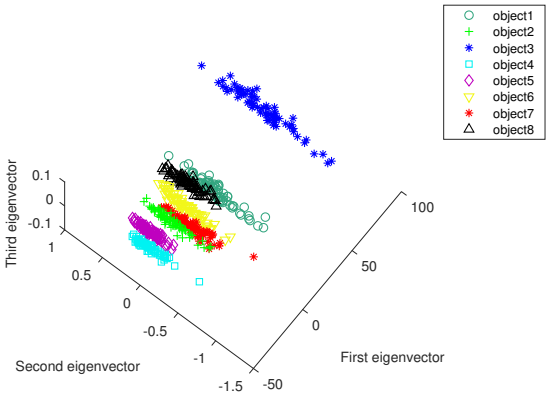
(c) $s = 3$

(d) $s = 4$

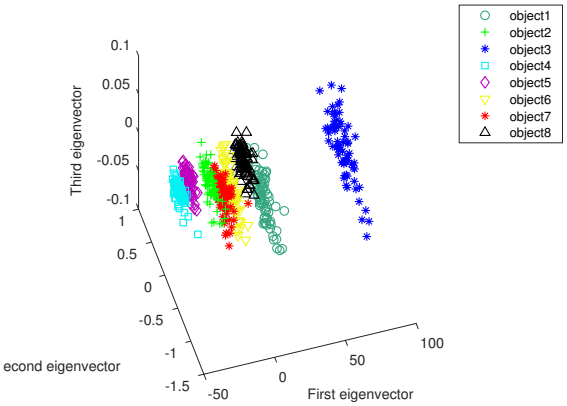
Figure 4.30: Illustration of PCA based clustering for 8 selected objects of the COIL-100 database. The feature vector consist of heat content polynomial coefficients q_0, q_1, q_2, q_3. From left to right and top to bottom, plots correspond to Mellin exponents $s = 1$, 2, 3, and 4.



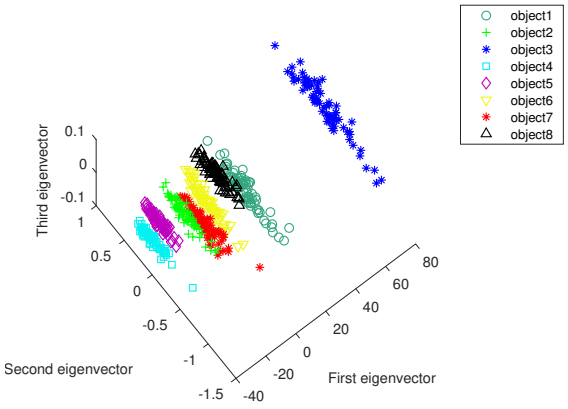
(a) $\lambda = 1$



(b) $\lambda = 2$



(c) $\lambda = 3$



(d) $\lambda = 4$

Figure 4.31: Illustration of PCA based clustering for 8 selected objects of the COIL-100 database. The feature vector consist of heat content polynomial coefficients q_0, q_1, q_2, q_3. From left to right and top to bottom, plots correspond to Laplace exponents $\lambda = 1$, 2, 3, and 4.

4.13 Conclusion

In this chapter, we have reviewed the heat kernel of a graph based on the standard diffusion model as discussed in [118] . We extended this concept by introducing the heat kernel (which we refer to as the generalised heat kernel) obtained as a fundamental solution to the heat equation where diffusion occurs through both direct (along edges) and long-range interactions. The latter are accounted for using either the Mellin or Laplace transforms of the k-path Laplacian matrices [38]. We have also discussed the utility of invariants of the heat kernel namely the heat kernel trace, zeta function, derivative of the zeta function at the origin, and heat content invariants for characterisation of graphs as put forward by Xiao [118]. We further showed that the same set of invariants of the generalised heat kernel can as well be used for graph characterisation for purposes of clustering as depicted by the results of experiments performed on selected objects of the COIL database. We also introduced the Laplacian energy

computed using the eigenvalues of both the standard and the generalised Laplacian matrix as a parameter independent feature for characterisation of graphs.

We have also investigated the impact of long-range interactions to object clustering. This is done by varying the values of the Mellin and Laplace exponents, s and λ respectively. The results from the experiments are two fold: For analysis in which feature vectors are constructed from the zeta function values for different values of p, it is observed that as s or λ increases, resulting clusters become more overlapped. On the other hand, using the derivative of the zeta function at the origin, better clusters are obtained as s or λ increases.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Diffusion is one of the dynamical processes that occurs on networks. It is of great importance as it aids in developing models of processes on real-world networks. Such processes include spread of epidemics in populations, dissemination of information in social networks, propagation of viruses in computer networks, among others. Our concern in this thesis lies in exploring an important aspect of diffusion which is the heat kernel and its utility in graph characterisation for purposes of clustering. Before diving into the study of the heat kernel, it was necessary first to understand in detail the diffusion process on (undirected) networks and this we did by discussing in some detail the standard diffusion model characterised by the flow of substance only along the edges of the network. We further performed simulations depicting the equilibrium behaviour under this model, and the impact of factors such as network structure, choice of source nodes and network heterogeneity on the rate of diffusion on simple undirected networks. Based on empirical evidence, Estrada [38] put forward the fact that in many dynamical processes, such as consensus, synchronisation, diffusion dynamics, among others, interaction among constituent components does not only occur among nearest neighbours but also between non-nearest neighbours which are referred to as long-range interactions. Long-range interactions have been proven to play a key role in many dynamic processes.

We thus turned our focus to studying the diffusion model (which we referred to as the generalised diffusion model) in which the flow of substance (particularly heat in this thesis) does not occur only along edges but also through long-range interactions which are accounted for using the Mellin and Laplace transforms of the k-path Laplacian matrices as introduced by Estrada [38]. Through simulations, we showed that diffusion is accelerated when long-range interactions are considered. Moreover, using simulation on artificial networks, we showed that the impact of long-range interactions is more pronounced in networks that follow power law degree distribution (BA networks) than in networks with Poisson degree distribution (ER networks). This is because of the existence of hubs in the former and node degree homogene-

ity in the latter.

The heat kernel is the fundamental solution of the diffusion equation. It describes the flow of substance over a network. In [118], the standard heat kernel is defined in such a way that it describes flow of substance only along edges of the network. Furthermore, Xiao extensively discussed the use of the standard heat kernel invariants such as the trace of the heat kernel, zeta function, derivative of the zeta function and heat content for characterisation of graphs and application to image analysis. In this thesis, we extended this concept by considering the heat kernel obtained as a solution to heat diffusion equations in which long-range interactions are taken into consideration. This heat kernel, which we refer to as the generalised heat kernel, is associated with the eigenvalues of the generalised Laplacian matrix. We further explored the use of invariants of the generalised heat kernel for characterisation of graphs and how stable feature vectors based on these invariants can be extracted and used in graph clustering. These invariants include the heat kernel, zeta function, derivative of the zeta function at the origin, and coefficients of the heat kernel. Similar to the standard heat kernel invariants, the above mentioned invariants associated with the generalised heat kernel do perform well too in distinguishing of graphs.

For object clustering, we performed experiments using 8 selected objects of the COIL database where we constructed feature vectors using the mentioned invariant and then performed principal component analysis on the dataset composed of the feature vectors. Results from the experiments showed that the invariants of the generalised heat kernel perform well in characterising graphs as is the case with the invariants of the standard heat kernel discussed in [118]. For comparison, we observed that clusters resulting from the use of the coefficients of the heat content are quite separated, that is to say, the overlap between clusters is minimal. This then followed by the zeta function whose associated clusters demonstrate less overlap compared to clusters resulting from the use of feature vector composed of the leading eigenvalues of the Laplace matrix.

5.2 Contributions

In summary, our contributions to this area include the following: we have introduced the generalised heat kernel which describes the flow of substance along the edges of the network and through long-range interactions. We have shown that invariants, namely the trace, zeta function, heat content, Laplacian energy and the derivative of the zeta function at the origin, associated with the generalised heat kernel can be used for characterisation of graphs and clustering of objects thus extending the approach of Xiao [118, 119] to the generalised heat kernel case as depicted by results of experiments in chapter 4. We also explored the impact of long-range influence on the quality of object clusters. Results of experiments of object clustering reveal that, for feature vectors constructed from the values of the zeta function at

different values of argument,p, as the influence due to long-range interaction weakens (by an increase in the Mellin or Laplace exponents (s or λ), the resulting clusters become more overlapped and as long-range interactions become stronger, the resulting clusters tend to be less overlapped implying better clusters. On the other hand, object clustering based on the derivative of the zeta function at the origin show an opposite trend to the previous case, that is to say, as the influence in long-range increases, better clusters are obtained. For clustering based on coefficients of the heat content polynomials, the variation in long-range influence results into slight variation in the quality of clusters.

5.3 Directions for future research

We have discussed the generalised diffusion model in which both direct and long-range interactions are considered. To account for long-range interactions, we use the method based on the Mellin and Laplace transforms of the k-path Laplacian matrices introduced by Estrada [38]. In this method, the strength of long-range interactions between pairs of nodes decreases as the shortest distance between the nodes increases.

The heat kernel of a diffusion equation is a very key concept in diffusion on networks and has various applications such as page rank measure of a graph [24], graph characterisation and clustering [118, 119], among others. In particular, Xiao [118] extensively studied the use of the standard heat kernel invariants namely the heat kernel trace, zeta function, derivative of zeta function at origin and the heat content for graph characterisation. Using feature vectors constructed using these invariants, Xiao performed principal component analysis on selected objects of the COIL database to obtain different object clusters.

The long-range interactions have proven significant in accelerating dynamical processes such as consensus [38], synchronisation [39] and diffusion on networks [40]. Focusing on diffusion on network with long-range interactions, we have explored the heat kernel resulting from this diffusion model and its applicability to graph characterisation for clustering purposes. Using experiments on 8 selected objects of the COIL database, we have shown that the invariants of the generalised heat kernel namely the trace, zeta function, derivative of the zeta function and heat content, do perform well in object clustering. Most interesting is the fact that as we vary the values of the Mellin transform or Laplace transform exponents (s or λ), we obtain an ensemble of clusters. Important to note is that as the values of exponents s or λ increase, the resulting clusters are less overlapped and this poses the following open question: at what value or range of values of the exponents (s or λ) do we obtain meaningful clusters? We hope our curiosity will open more doors to interesting research...

List of References

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

[2] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.

[3] Michael O Albertson. The irregularity of a graph. *Ars Combinatoria*, 46:219–225, 1997.

[4] NOBLE Alison. *Descriptions of Image Surfaces*. PhD thesis, Department of Engineering Science, Oxford University, 1989: 45, 1989.

[5] Uri Alon. Biological networks: the tinkerer as an engineer. *Science*, 301(5641):1866–1867, 2003.

[6] Luıs A Nunes Amaral, Antonio Scala, Marc Barthelemy, and H Eugene Stanley. Classes of small-world networks. *Proceedings of the national academy of sciences*, 97(21):11149–11152, 2000.

[7] Howard Anton and Chris Rorres. Elementary linear algebra, (2000). *Anton Textbook Inc, Ottawa*, 2007.

[8] Franz Aurenhammer and Rolf Klein. Voronoi diagrams. *Handbook of computational geometry*, 5:201–290, 2000.

[9] Jayanth R Banavar, Amos Maritan, and Andrea Rinaldo. Size and form in efficient transportation networks. *Nature*, 399(6732):130, 1999.

[10] Albert-László Barabási. Linked: The new science of networks, 2003.

[11] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[12] C Bradford Barber, David P Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)*, 22(4):469–483, 1996.

[13] Alain Barrat, Marc Barthelemy, and Alessandro Vespignani. *Dynamical processes on complex networks*. Cambridge university press, 2008.

[14] Francis K Bell. A note on the irregularity of graphs. *Linear Algebra and its Applications*, 161:45–54, 1992.

[15] Azer Bestavros. Discussion on Eulerian circuits. Online; accessed 2017-03-29.

[16] Norman Biggs. *Algebraic graph theory*. Cambridge university press, 1993.

[17] S Burcu Bozkurt, A Dilek Güngör, Ivan Gutman, and A Sinan Cevik. Randic matrix and Randic energy. *MATCH Commun. Math. Comput. Chem*, 64:239–250, 2010.

[18] Anna D Broido and Aaron Clauset. Scale-free networks are rare. *arXiv preprint arXiv:1801.03400*, 2018.

[19] Andries E Brouwer and Willem H Haemers. *Spectra of graphs*. Springer Science & Business Media, 2011.

[20] Alexander Crum Brown and Thomas R Fraser. On the connection between chemical constitution and physiological action; with special reference to the physiological action of the salts of the ammonium bases derived from strychnia, brucia, thebaia, codeia, morphia, and nicotia. *Journal of anatomy and physiology*, 2(2):224, 1868.

[21] Horst Bunke. Error correcting graph matching: On the influence of the underlying cost function. *IEEE transactions on pattern analysis and machine intelligence*, 21(9):917–922, 1999.

[22] John L. Casti. MS Windows NT complexity, September 26, 2017.

[23] Boris V Cherkassky, Andrew V Goldberg, and Tomasz Radzik. Shortest paths algorithms: Theory and experimental evaluation. *Mathematical programming*, 73(2):129–174, 1996.

[24] Fan Chung. The heat kernel as the pagerank of a graph. *Proceedings of the National Academy of Sciences*, 104(50):19735–19740, 2007.

[25] Fan Chung. A local graph partitioning algorithm using heat kernel pagerank. *Internet Mathematics*, 6(3):315–330, 2009.

[26] Fan RK Chung. *Spectral graph theory*. Number 92. American Mathematical Soc., 1997.

[27] David Cohen. All the world's a net. *New Scientist*, 174(2338):24–9, 2002.

[28] Ethan Cohen-Cole and Jason M Fletcher. Is obesity contagious? social networks vs. environmental factors in the obesity epidemic. *Journal of health economics*, 27(5):1382–1387, 2008.

[29] DM Cvetkovic and PETER Rowlinson. Spectral graph theory. *Topics in algebraic graph theory*, pages 88–112, 2004.

[30] K Ch Das. The Laplacian spectrum of a graph. *Computers & Mathematics with Applications*, 48(5):715–724, 2004.

[31] Thomas Deselaers, Daniel Keysers, and Hermann Ney. Features for image retrieval: an experimental comparison. *Information retrieval*, 11(2):77–107, 2008.

[32] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[33] Simena Dinas and José María Banon. A review on Delaunay triangulation with application on computer vision. *Int. J. Comp. Sci. Eng*, 3:9–18, 2014.

[34] Chris Ding and Xiaofeng He. K-means clustering via principal component analysis. In *Proceedings of the twenty-first international conference on Machine learning*, page 29. ACM, 2004.

[35] Paul Erdös and Alfréd Rényi. On random graphs. *Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959.

[36] Ernesto Estrada. Quantifying network heterogeneity. *Physical Review E*, 82(6):066102, 2010.

[37] Ernesto Estrada. *The structure of complex networks: theory and applications*. OUP Oxford, 2011.

[38] Ernesto Estrada. Path Laplacian matrices: introduction and application to the analysis of consensus in networks. *Linear Algebra and its Applications*, 436(9):3373–3391, 2012.

[39] Ernesto Estrada, Lucia Valentina Gambuzza, and Mattia Frasca. Long-range interactions and network synchronization. *arXiv preprint arXiv:1704.01349*, 2017.

[40] Ernesto Estrada, Ehsan Hameed, Naomichi Hatano, and Matthias Langer. Path Laplacian operators and superdiffusive processes ongraphs. i. one-dimensional case. *Linear Algebra and its Applications*, 523:307–334, 2017.

[41] Ernesto Estrada, Franck Kalala-Mutombo, and Alba Valverde-Colmeiro. Epidemic spreading in networks with nonrandom long-range interactions. *Physical Review E*, 84(3):036110, 2011.

[42] Ernesto Estrada, Philip Knight, et al. *A first course in network theory*. Oxford University Press, USA, 2015.

[43] L Euler. The solution of a problem relating to the geometry of position. 1976.

[44] Leonhard Euler. Leonhard euler and the Königsberg bridges. *Scientific American*, 189(1):66–70, 1953.

[45] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *ACM SIGCOMM computer communication review*, volume 29, pages 251–262. ACM, 1999.

[46] Steven Fortune. Voronoi diagrams and Delaunay triangulations. In *Computing in Euclidean geometry*, pages 225–265. World Scientific, 1995.

[47] Fabien Friedli, Anders Karlsson, et al. Spectral zeta functions of graphs and the Riemann zeta function in the critical strip. *Tohoku Mathematical Journal*, 69(4):585–610, 2017.

[48] Xinbo Gao, Bing Xiao, Dacheng Tao, and Xuelong Li. A survey of graph edit distance. *Pattern Analysis and applications*, 13(1):113–129, 2010.

[49] Chris Godsil and G Royle. Algebraic graph theory springer. *New York*, 2001.

[50] Leo Grady. Random walks for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 28(11):1768–1783, 2006.

[51] Richard Green. Zachary's karate club network, 2012. [Online; accessed 2017-04-25].

[52] Ivan Gutman. The energy of a graph: old and new results. In *Algebraic combinatorics and applications*, pages 196–211. Springer, 2001.

[53] Ivan Gutman and Oskar E Polansky. *Mathematical concepts in organic chemistry*. Springer Science & Business Media, 2012.

[54] Ivan Gutman and Bo Zhou. Laplacian energy of a graph. *Linear Algebra and its applications*, 414(1):29–37, 2006.

[55] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.

[56] Zaïd Harchaoui and Francis Bach. Image classification with segmentation graph kernels. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[57] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988.

[58] John Michael Harris, Jeffry L Hirst, and Michael J Mossinghoff. *Combinatorics and graph theory*, volume 2. Springer, 2008.

[59] Stephen Hawking. San Jose Mercury News, 2000.

[60] Xiaofei He and Partha Niyogi. Locality preserving projections. In *Advances in neural information processing systems*, pages 153–160, 2004.

[61] Bernardo A Huberman. The laws of the web, 2001.

[62] Matthew O Jackson. *Social and economic networks*. Princeton university press, 2010.

[63] Rinku Jacob, KP Harikrishnan, R Misra, and G Ambika. Measure for degree heterogeneity in complex networks and its application to recurrence network analysis. *Royal Society open science*, 4(1):160757, 2017.

[64] Barbara R Jasny and L Bryan Ray. Life and the art of networks, 2003.

[65] Lili Ju, Todd Ringler, and Max Gunzburger. Voronoi tessellations and their application to climate and global modeling. In *Numerical techniques for global atmospheric models*, pages 313–342. Springer, 2011.

[66] Abraham Kandel, Horst Bunke, and Mark Last. *Applied graph theory in computer vision and pattern recognition*, volume 52. Springer, 2007.

[67] Jieqi Kang. Image processing and understanding based on graph similarity testing: algorithm design and software development. 2017.

[68] Michał Karoński. A review of random graphs. *Journal of Graph Theory*, 6(4):349–389, 1982.

[69] R Kasprzak. Diffusion in networks. *Journal of Telecommunications and Information Technology*, pages 99–106, 2012.

[70] Erica Klarreich. Scant evidence of power laws found in real- world networks, February 15, 2018.

[71] Jon M Kleinberg. Navigation in a small world. *Nature*, 406(6798):845, 2000.

[72] Kyle Kloster and David F Gleich. Heat kernel based community detection. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1386–1395. ACM, 2014.

[73] Oliver Knill. The zeta function for circular graphs. *arXiv preprint arXiv:1312.4239*, 2013.

[74] Ewa Kubicka, Grzegorz Kubicki, and Ignatios Vakalis. Using graph distance in object recognition. In *Proceedings of the 1990 ACM annual conference on Cooperation*, pages 43–48. ACM, 1990.

[75] Giorgio Levi. A note on the derivation of maximal common subgraphs of two directed or undirected graphs. *Calcolo*, 9(4):341, 1973.

[76] Dunia López-Pintado. Diffusion in complex social networks. *Games and Economic Behavior*, 62(2):573–590, 2008.

[77] Bin Luo, Antonio Robles-Kelly, Andrea Torsello, Richard C Wilson, and Edwin R Hancock. Discovering shape categories by clustering shock trees. In *International Conference on Computer Analysis of Images and Patterns*, pages 152–160. Springer, 2001.

[78] Bin Luo, Richard C Wilson, and Edwin R Hancock. Spectral clustering of graphs. In *International Workshop on Graph-Based Representations in Pattern Recognition*, pages 190–201. Springer, 2003.

[79] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. Mining social networks using heat diffusion processes for marketing candidates selection. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 233–242. ACM, 2008.

[80] Justin Magouirk, Scott Atran, and Marc Sageman. Connecting terrorist networks. *Studies in Conflict & Terrorism*, 31(1):1–16, 2008.

[81] Patrick McDonald and Robert Meyers. Diffusions on graphs, Poisson problems and spectral geometry. *Transactions of the American Mathematical Society*, 354(12):5111–5136, 2002.

[82] Sergey Melnik, Hector Garcia-Molina, and Erhard Rahm. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *Data Engineering, 2002. Proceedings. 18th International Conference on*, pages 117–128. IEEE, 2002.

[83] Stanley Milgram. The small world problem. *Psychology today*, 2(1):60–67, 1967.

[84] Hans P Moravec. Visual mapping by a robot rover. In *Proceedings of the 6th international joint conference on Artificial intelligence-Volume 1*, pages 598–600. Morgan Kaufmann Publishers Inc., 1979.

[85] Sameer A Nene, Shree K Nayar, Hiroshi Murase, et al. Columbia object image library (coil-100). 1996.

[86] Mark Newman. *Networks: an introduction*. OUP Oxford, 2010.

[87] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[88] Mark EJ Newman, Duncan J Watts, and Steven H Strogatz. Random graph models of social networks. *Proceedings of the National Academy of Sciences*, 99(suppl 1):2566–2572, 2002.

[89] Mladen Nikolić. Measuring similarity of graph nodes by neighbor matching. *Intelligent Data Analysis*, 16(6):865–878, 2012.

[90] Computer Systems Odessa. Computer network diagrams. [copyright 1993-2018 CS Odessa Corp].

[91] Atsuyuki Okabe, Barry Boots, Kokichi Sugihara, and Sung Nok Chiu. *Spatial tessellations: concepts and applications of Voronoi diagrams*, volume 501. John Wiley & Sons, 2009.

[92] Giuliano Andrea Pagani and Marco Aiello. The power grid as a complex network: a survey. *Physica A: Statistical Mechanics and its Applications*, 392(11):2688–2700, 2013.

[93] Seth Pettie. A faster all-pairs shortest path algorithm for real-weighted sparse graphs. In *International Colloquium on Automata, Languages, and Programming*, pages 85–97. Springer, 2002.

[94] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European conference on computer vision*, pages 430–443. Springer, 2006.

[95] Benno Schwikowski, Peter Uetz, and Stanley Fields. A network of protein–protein interactions in yeast. *Nature biotechnology*, 18(12):1257, 2000.

[96] Raimund Seidel. On the all-pairs-shortest-path problem in unweighted undirected graphs. *Journal of computer and system sciences*, 51(3):400–403, 1995.

[97] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[98] Jonathon Shlens. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*, 2014.

[99] Ali Shokoufandeh, Sven J Dickinson, Kaleem Siddiqi, and Steven W Zucker. Indexing using a spectral encoding of topological structure. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2, pages 491–497. IEEE, 1999.

[100] Danuta M Skowronski, Caroline Astell, Robert C Brunham, Donald E Low, Martin Petric, Rachel L Roper, Pierre J Talbot, Theresa Tam, and Lorne Babiuk. Severe acute respiratory syndrome (sars): a year in review. *Annu. Rev. Med.*, 56:357–381, 2005.

[101] Olaf Sporns, Dante R Chialvo, Marcus Kaiser, and Claus C Hilgetag. Organization, development and function of complex brain networks. *Trends in cognitive sciences*, 8(9):418–425, 2004.

[102] Adina Raluca Stoica. Delaunay diagram representations for use in image near-duplicate detection. *Senior project submitted to the division of science, mathematics and computing of Bard College. New York*, 2011.

[103] Reiko Tanaka. Scale-rich metabolic networks. *Physical review letters*, 94(16):168101, 2005.

[104] Warren S Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419, 1952.

[105] Andrea Torsello, Antonio Robles-Kelly, and Edwin R Hancock. Discovering shape classes using tree edit-distance and pairwise clustering. *International Journal of Computer Vision*, 72(3):259–285, 2007.

[106] Miroslav Trajković and Mark Hedley. Fast corner detection. *Image and vision computing*, 16(2):75–87, 1998.

[107] Alexander Tsiatas. *Diffusion and clustering on large graphs*. University of California, San Diego, 2012.

[108] György Turán. On the succinct representation of graphs. *Discrete Applied Mathematics*, 8(3):289–294, 1984.

[109] Julian R Ullmann. An algorithm for subgraph isomorphism. *Journal of the ACM (JACM)*, 23(1):31–42, 1976.

[110] Mauricio Uriona and Caroline R Vaz. The evolution of sustainability transitions and technological innovation systems research: a bibliometric analysis.

[111] Fabien Viger and Matthieu Latapy. Efficient and simple generation of random simple connected graphs with prescribed degree sequence. In *International Computing and Combinatorics Conference*, pages 440–449. Springer, 2005.

[112] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world' networks. *nature*, 393(6684):440–442, 1998.

[113] Louis Weinberg. A simple and efficient algorithm for determining isomorphism of planar triply connected graphs. *IEEE Transactions on Circuit Theory*, 13(2):142–148, 1966.

[114] Barry Wellman and Stephen D Berkowitz. *Social structures: A network approach*, volume 2. CUP Archive, 1988.

[115] Tina Wey, Daniel T Blumstein, Weiwei Shen, and Ferenc Jordan. Social network analysis of animal behaviour: a promising tool for the study of sociality. *Animal behaviour*, 75(2):333–344, 2008.

[116] Richard C Wilson, Edwin R Hancock, and Bin Luo. Pattern vectors from algebraic graph theory. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1112–1124, 2005.

[117] Robin J Wilson. *An introduction to graph theory*. Pearson Education India, 1970.

[118] Bai Xiao. *Heat Kernel Analysis on Graphs*. PhD thesis, The University of York, 2007.

[119] Bai Xiao, Edwin R Hancock, and Richard C Wilson. Graph characteristics from the heat kernel trace. *Pattern Recognition*, 42(11):2589–2606, 2009.

[120] Laura A Zager and George C Verghese. Graph similarity scoring and matching. *Applied mathematics letters*, 21(1):86–94, 2008.